Copyright

by

Seokyoung Ahn

2005

The Dissertation Committee for Seokyoung Ahn certifies that this is the approved version of the following dissertation:

Modeling, Estimation, and Control of Electroslag Remelting Process

Committee:

Joseph J. Beaman, Supervisor

Robert H. Bishop

David L. Bourell

John R. Howell

Raul G. Longoria

Modeling, Estimation, and Control of Electroslag Remelting Process

by

Seokyoung Ahn, B.E., M.S.

Dissertation

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

The University of Texas at Austin
May 2005

Dedication

To my lovely wife Young-Seol

Acknowledgements

First of all, I would like to thank my advisor, Dr. Joseph J. Beaman, whose patient support and enduring confidence have helped to guide me throughout my work and time at Austin. I thank Professor Robert H. Bishop for providing valuable advice and fundamental guidance throughout my doctoral tenure. I thank Dr. David L. Bourell, Dr. John R. Howell, and Dr. Raul G. Longoria for guiding my work as members of my dissertation committee.

I would also like to acknowledge Cindy Pflughoft for her kindness and magical support during my research. Additional gratitude also must go to my colleagues, Daphne, Carey, Seok-Min, Jeremy, Jorge, Aparajit, Ravi, Ji-hoon, Moss, Srikanth, Vinay, Mario, Amol, and Dr. Rhee.

This work could not have been completed without my wife Young-Seol, who was patient and sincerely supportive. Finally, I am always indebted to my parents and parents-in-law for the love, support, and patience they show me.

Modeling, Estimation, and Control of Electroslag Remelting Process

Publication No._____________

Seokyoung Ahn, PhD.
The University of Texas at Austin, 2005

Supervisor: Joseph J. Beaman

Electroslag Remelting (ESR) is used widely throughout the specialty metals industry to produce superalloy and special steel cast ingots. High quality ESR casting requires that the electrode melting rate be controlled at all times during the process. This is especially difficult when process conditions are such that the temperature distribution in the electrode has not achieved, or has been driven away from, steady state. This condition is encountered during the beginning and closing stages of the ESR process and also during some process disturbances such as when the melt zone passes through a transverse crack. To address these transient melting situations, a new method of ESR melt rate control has been developed that incorporates an accurate, reduced-order melting model to continually estimate the temperature distribution in the electrode. The related state variables are estimated by the observer algorithms. Due to the highly nonlinear characteristics of the process, more sophisticated estimators than the Kalman filter are proposed. The unscented Kalman filter (UKF) based on the unscented transform and the particle filtering technique were chosen for possible candidates and applied in the controller design. During the highly transient periods during melting, the UKF showed the best performance for controlling the melt rate. Particle filtering can deal with non-Gaussian noises and the accuracy is totally based on the number of the Monte Carlo runs. Unfortunately, the particle filter is relatively slow in the real-time applications for controlling the ESR process with current computer technology.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1 DESCRIPTION OF THE ELECTROSLAG REMELTING PROCESS

Electroslag Remelting (ESR) is a secondary melting process used to produce a variety of metals and alloys (e.g. premium-grade superalloys, stainless steels, and high-strength or high-temperature alloys). The ESR process is widely used among producers of specialty metals and alloys because it is capable of producing very clean materials with absence of defects such as pipe and porosity. Remelting is utilized to produce fully dense, defect-free, homogeneous ingots with an appropriate chemistry, physical size, and grain structure. Research is performed on ESR and other furnace remelting techniques through the Specialty Metals Processing Consortium in conjunction with several national laboratories [32]. See Figure 1.1 for a traditional commercial ESR setup.

As shown in the Figure 1.2, a typical furnace is composed of a consumable electrode and a water-cooled crucible. The process starts from the bottom of the crucible. An arc is struck between the bottom starting plate and the electrode which begins to melt powdered slag placed in the crucible. Once the slag melts, it becomes electrically conductive, and the tip of the electrode immersed in the slag starts to melt. As the electrode melts, droplets of metal fall through the slag and are refined by contact with the slag under mechanical, thermochemical and electrochemical mechanisms [42]. The droplets collect at the bottom of the water-cooled crucible and solidify to form an ingot with low residual impurities, very few non-metallic inclusions, and excellent surface quality.

Figure 1.1 20 ton ESR furnace capable of melting under protective atmosphere (courtesy of http://www.ald-vt.com)

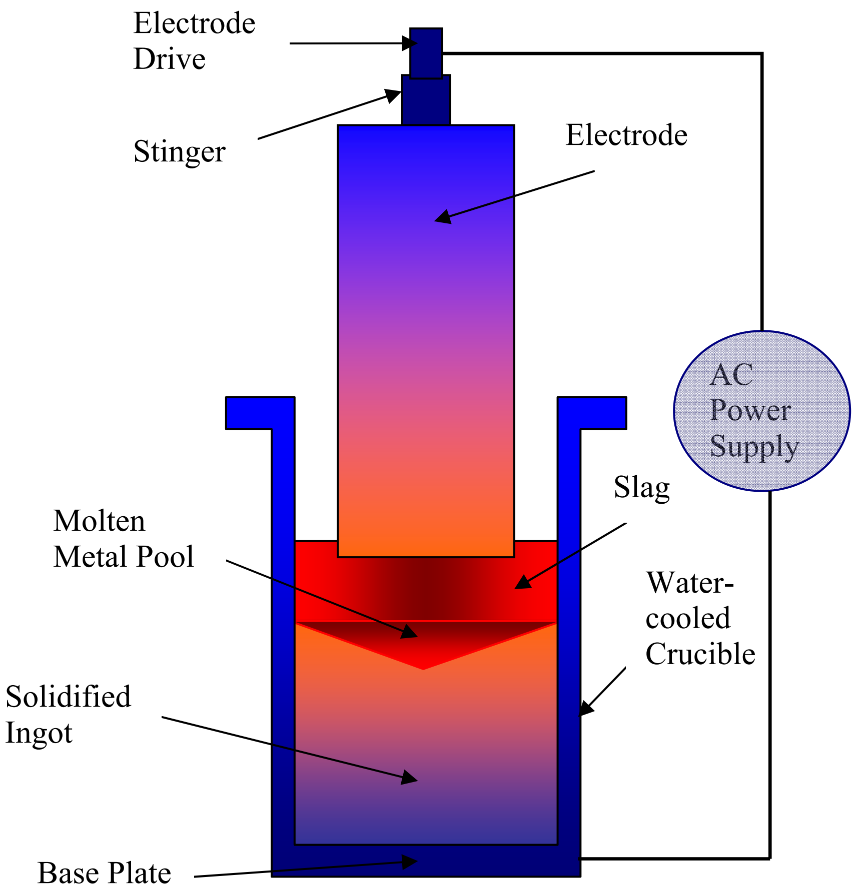


Figure 1.2 ESR process description

During the process, the top of the ingot remains a pool of liquid metal that has a “V” shaped cross section. The molten slag floats on top of the liquid metal. As more of the electrode melts and falls through the slag, the ingot grows upwards and hence the slag pool also rises. Typically an alternating current (AC) is passed through the electrode into the slag and then out at the base plate, although sometimes the mold itself can carry current [43].

The slag used in the process is a mixture of ceramics and other ingredients that are electrically conductive when molten. Some typical primary ingredients are CaF_2, Al_2O_3, MgO, or CaO. The chemistry of the slag may be determined by eutectic points or slag properties, such as vapor pressure, melting points, resistivity, viscosity, density, specific heat, and surface tension or to promote particular chemical reactions [32]. Because of the potential reactions between the electrode and the slag, ESR is uniquely capable of removing inclusions and altering the original electrode composition to meet the chemical specifications for a particular alloy [32]. The slag is used in the process to release heat to melt the electrode, to purify and protect the liquid metal, and to create a mold lining. The only significant energy used to melt the electrode is Joule heating which occurs as the current is passed through the slag. The temperature of the molten slag is usually about 200 to 300 °C higher than the melting temperature of the electrode. As the electrode tip is immersed in the slag, it melts and drops through the slag because the molten metal is significantly denser. Some slag also freezes against the mold wall, which prevents direct contact between the molten metal and the copper crucible. The frozen slag is normally referred to as the slag skin. This skin serves as an electrical and thermal insulator.

The electrode is usually welded to a stub, called a stinger that is attached to a linear actuator. The linear actuator is a part of the electrical circuit with one or more load cells to measure the electrode weight. Since the crucible diameter is larger than the electrode diameter, the electrode must be lowered into the slag as the electrode tip melts to maintain a constant immersion depth. Immersion depth of the electrode in the slag has an important effect on the heat released during the process. Since the slag acts as a resistor, the deeper the electrode is place into the slag, the lower the resistance and the lower the thermal input into the electrode and molten slag. Conversely, lower immersion depths result in higher slag resistance and higher the thermal output to the electrode and slag. If the thermal input decreases too much, the slag can begin to cool resulting in a rough ingot surface or a slag skin rupture where the molten metal breaks through the slag surface. Therefore in general, the electrode is kept as close to the slag surface as possible to keep the energy input into the slag high, preventing slag cooling and maintaining good ingot surface characteristics.

Another important aspect of ESR is the prevention of segregation. Because of the water-cooled copper crucible, the ingot tends to undergo directional solidification which will ideally yield an ingot without segregation in the material. Crucibles are often made of copper, but other materials such as aluminum and steel can be used. Many molds are round but they can also be square or even have a center to create tube shaped ingots [43]. Coolant is passed through the mold or in the surface of the mold using a liner. Some crucibles are longer that the length of the ingot and remain stationary, while other molds move with the growing ingot and are much shorter than the stationary molds.

1.2 MOTIVATION FOR MODELING, ESTIMATION, AND CONTROL OF ELECTROSLAG REMELTING PROCESS

Stable, responsive control of the electroslag remelting is critical for manufacturing uniform, defect-free ingots especially for super alloy applications. Current ESR control systems are closed loop, single input-single output (SISO) systems that generalize the physical processes involved in ESR. The increasing demand for cleaner, more highly engineered, chemically tuned alloys has pushed the current control techniques to their limits [32]. The current SISO control methods fail to produce acceptable quality ingots and hinder producers in their efforts to raise quality standards because of the complexity of the actual ESR process [42]. The Specialty Metals Processing Consortium (SMPC) in conjunction with Sandia National Laboratories has identified the need for new ESR control methods [32]. Extensive research into reduced-order models of remelting techniques and the controller design has been done by Joseph J. Beaman at the University of Texas at Austin [4, 5, 6, 49, 50, 51]. Another popular remelting technique, vacuum arc remelting (VAR), has been researched and a novel model-based control design was implemented and proved to be useful in the industry [4, 5, 6, 8, 49]. ESR and VAR share many similarities in modeling and control design. Modeling of the ESR in the next chapter is based on the model of the VAR derived by Beaman et al [5].

The first step in developing an accurate control method is deriving an energy-based model that properly describes the physical phenomena occurring during the ESR process. For example, to control the melt rate it is necessary to determine what parameters of the ESR process affect the melt rate. The goal is to use the energy-based modeling approach to model the ESR process while including all the important process parameters. From Chapter 2, an energy-based reduced order ESR model was developed, simulated, and experimentally verified at Sandia National Laboratories in Albuquerque, New Mexico.

Open loop approaches for the control are not robust with respect to changes in the process condition. Robust feedback control of the melt rate and the immersion depth in the ESR process requires knowledge of the melting conditions through measurements, which are then incorporated into the overall feedback control system. As in Beaman's approach for controlling the VAR process [6], we use the observer-based control design for ESR. To use feedback control we need an estimator because several state variables cannot directly be measured or are immeasurable. For VAR, a linearized time-invariant Kalman filter algorithm has been used to design an observer. For this case, the Kalman gain is calculated off-line so it is a constant matrix based on the nominal process values. For steady state, estimation by constant gain Kalman filter is acceptable, but for a highly transient processes during the ESR process such as start-up or hot-top periods, we need a more sophisticated approach for estimating the related variables because the performance of the suggested control method using causality reasoning of the dynamics depends on the quality of estimation. For nonlinear dynamics problems, the extended Kalman filter has been widely used and accepted for several decades [22, 31, 14]. However, due to the first order Taylor series approximation of the nonlinearities, for highly nonlinear problems, more accurate filters are needed and have been researched. Reviewing all the work would be very lengthy and is beyond the scope of this dissertation.

Recently, Julier and Uhlmann proposed the use of a novel unscented transformation to approximate the statistics of the nonlinear transformation with carefully chosen sample points rather than approximating the nonlinear model itself with truncated Taylor series and applied it to the Kalman filter framework [23, 25]. The mean estimation of the unscented Kalman filter has been shown to be correct up to the second order of its Taylor series expansion [27]. Also the unscented Kalman filter does not require the calculation of the Jacobians which is a definite benefit for complex dynamics where the derivatives are not defined properly. From the literature, the unscented Kalman filter performs better than the extended Kalman filter in most cases [9, 26, 48]. The idea using the deterministic sampling method has been researched independently from other groups, too. Nørgaard et al proposed the divided difference filter which uses polynomial interpolations of the nonlinear transformation by replacing the analytical Jacobian of the extended Kalman filter with numerically evaluated divided differences [35]. Ito and Xiong introduced a similar version named as the central difference filter and state that the filter performs better than or as good as the unscented Kalman filter [21]. The members of this class of filters differ only in the method of selecting the sample points, its number, value, and weights. But the deterministic sampling based probability distribution approximation filter such as the unscented Kalman filter still assumes a Gaussian distribution for the posterior density distribution. For a linear system, a Gaussian input guarantees a Gaussian output, but for a general nonlinear system this is not true. To resolve this issue, Monte Carlo-based recursive Bayesian approaches have been suggested. After the seminal paper by Gordon regarding the resampling technique [16], many variants of the particle filter were introduced [12]. It uses sequential Monte Carlo estimation based on a point mass or particle representation of the posterior distribution without making any assumptions to its shape; thus, it can be used with general nonlinear, non-Gaussian systems. Originally, the direct use of Monte Carlo techniques was proposed in 1960's and early 1970's [17, 18]. The computing power at the time wasn't enough to support the idea. Since the speed of computers has been exponentially increasing according to Moore's law, the use of the filtering algorithms based on the Monte Carlo technique continues to gain popularity.

We will review the filtering algorithms in Chapter 3 and Chapter 4. Finally, a linearzied time-invariant Kalman filter, extended Kalman filter, unscented Kalman filter, and a sequential Monte Carlo filter will be applied to the ESR control system and the performance will be compared at Chapter 5 and Chapter 6. Based on the performance comparison, the most appropriate filtering algorithm for the ESR process will be chosen.

The control problem for the ESR process and the goals of this Dissertation can be stated as follows:

Given desired values for the melt rate and immersion depth, determine the time varying current and ram speed inputs required to establish and maintain these desired outputs under the noisy environments.

The ESR control algorithm is based on inverting the normal causality of the dynamics and is presented in Chapter 5. The simulation results and discussion will be presented in Chapter 6.

1.3 THE CONTRIBUTION OF THIS DISSERTATION

Most of the objectives as stated in the previous section were successfully completed during the course of this Dissertation work. In summary, the following concrete and substantial contributions to the ESR process were made.

1. Modeling of the electroslag remelting process
    a. A model based on the energy relationship for ESR has been derived. The boundary layer approach and the approximate integral method were used to solve a Stefan moving boundary problem.
    b. For the measurement dynamics, the use of the impedance to determine the immersion depth has been formulated and implemented in the control design.
2. Design of the control algorithm based on the causality of the dynamics.
    a. A nonlinear controller based on the causality of the system dynamics has been suggested and implemented with integration of several observer algorithms.

3. Application of the suggested estimation algorithms to the control design of the ESR process.
    a. The linearized time-invariant Kalman filter has been applied to the control design.
    b. The extended Kalman filter has been applied to the control design.
    c. The Unscented Kalman filter has been applied to the control design.
    d. The sequential Monte Carlo filtering (particle filtering) technique based on the sampling importance resampling algorithm has been applied to the control design.
    e. Performance of the all of the filtering algorithms has been compared for the transient period (start-up and hot-top) and a traditional whole melt profile in practice of the ESR process.
4. Demonstration of the viability of using suggested robust control design of the ESR process.

CHAPTER 2

REDUCED ORDER DYNAMIC MODEL OF THE ELECTROSLAG REMELTING PROCESS

2.1 HEAT TRANSFER MODEL FOR ESR PROCESS

Transient heat conduction in a solid undergoing phase change represents a problem area of great technological importance. Examples of its application include metal casting, ice formation, space vehicle heat ablation, etc., to name only a few. Such problems are inherently nonlinear and involve a moving boundary whose location is usually unknown in advance. Exact solutions to these problems are very difficult and necessarily require considerable numerical computation, even if a simplified model of the problem is used.

Analytical approximation methods can be useful in providing approximate solutions to these problems and also a true insight into the physical phenomena can be obtained in the solution process. Discrete finite difference or finite element methods can be used with accurate results but the large number of coupled equations will make control design very difficult. In this chapter, we will discuss about the appropriate assumptions needed to apply the approximate solution to the ESR process, the reduced order model for the ESR process, and the additional dynamics such as buoyancy and immersion depth model.

A schematic heat transfer model is shown in Figure 2.1 and 2.2.

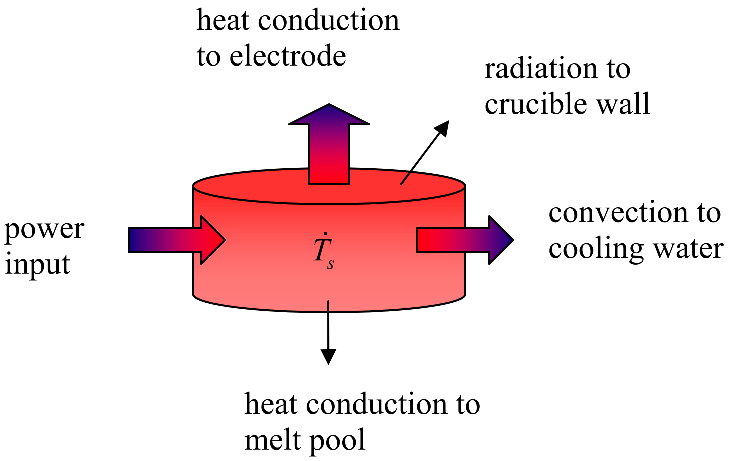


Figure 2.1 Energy balance of the slag in the ESR process

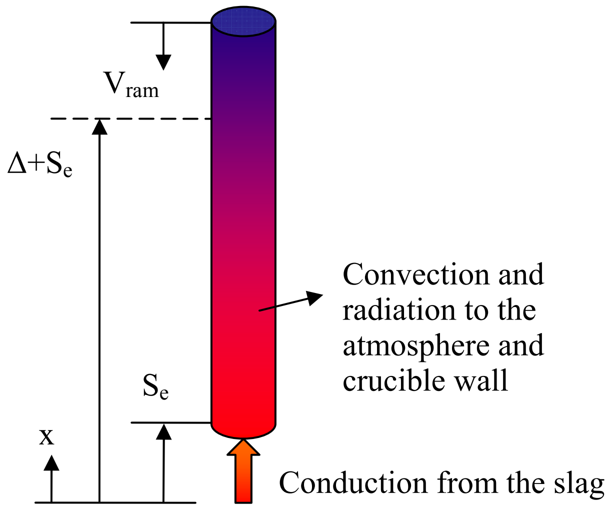


Figure 2.2 Energy balance and coordinate setup for the electrode

Heat to be generated in molten slag transfers from the boundaries of the molten slag to

1. the electrode surface immersed in it
2. the solid slag layer (slag skin) between the inner mold wall and the molten slag
3. the molten slag surface open to the atmosphere between the mold and electrode
4. the surface of the molten metal pool.

Based on the above heat transfer model, the following assumptions are made to derive the reduced order model for the ESR process.

1. The temperature distribution is almost uniform in the molten slag. The molten slag is observed to be stirring violently as observed in the molten metal pool and many authors assumed the uniform temperature distribution [4, 36]
2. The heat transfer coefficient between the molten slag and the slag skin is constant, and depends neither on the slag composition nor on the temperature of the molten slag
3. The electric current density distribution across the radius of the molten slag is uniform
4. Slag skin is at the uniform solidus temperature of slag (T_{ss})

5. Slag skin thickness, d_s and the crucible wall thickness, h_w satisfy $d_s \ll r_m$ and $h_w \ll r_m$ where r_m is radius of crucible
6. There is no direct heat transfer from the slag to the melt pool
7. Radiation occurs only between the slag surface and the crucible wall
8. The electrode conductivity in the slag K_e is described by $K_e = K_r e^{-C/T}$, where C is a constant and K_r is room temperature conductivity of the electrode
9. Slag skin can be treated as in equilibrium
10. The electrode face is flat throughout the process
11. Slag volume changes very slowly.

2.2 HEAT TRANSFER MODEL OF THE SLAG

The heat from the slag in the ESR process is dissipated through conduction to the electrode, and through convection to the coolant circulating in the crucible. Amounts of the energy dissipated by radiation are relatively small compared to the energy dissipation by conduction and convection [42]. The energy stored in the system is the difference between the energy coming in and the leaving the system. The energy coming into the system is the power generated from the current being run through the resistively heated slag. The slag layer is the element in the ESR furnaces with the greatest resistance and is responsible for the conversion of electrical energy to heat [13].

The energy balance in the ESR process is illustrated in Figure 2.1 and Figure 2.2. From the heat generated by the resistive heating in the slag, some is stored in the slag, some is dissipated by the convection to the cooling water in the crucible, and the rest is conducted to the electrode. The heat transfer between the slag and the pool is small compared to the convective losses to the cooling water, and the conduction to the electrode [36]. Therefore in the model, heat transfer between the slag and the melt pool is neglected. The thermal resistance of the crucible itself can be negligible because of the high thermal conductivity of the copper crucible walls [42]. The brief energy relationship of the power input and output for the slag is described as

$$\mathrm{P}_{\mathrm{out}} = \mathrm{P}_{\substack{\text{conduction}\\ \text{to electrode}}} + \mathrm{P}_{\substack{\text{convection}\\ \text{to cooling water}}} + \mathrm{P}_{\text{radiation}} \tag{2.1}$$

$$P_{out} = H_e A_e \left(T_s - T_m\right) + H_s 2\pi r_m h_s \left(T_s - T_{ss}\right) + \sigma\varepsilon\left(2\pi r_m^2 - 2\pi r_e^2\right)\left(T_s^4 - T_\infty^4\right) \tag{2.2}$$

where,

H_e : equivalent thermal coefficient of electrode (W/cm^{2}K)

A_e : area of cross section of the electrode (cm^2)

T_s : temperature of the slag (K)

T_{ss} : solidus temperature of the slag (K)

T_m : melt temperature of the electrode (K)

H_s : equivalent thermal coefficient of the slag (W/cm^{2}K)

r_e : radius of the electrode (cm)

r_m : radius of the crucible (cm)

h_s : height of the slag (cm)

σ : Stefan–Boltzmann constant (W/m^{2}K^4)

ε : emissivity of the slag

T_∞ : room temperature (K)

The stored energy inside the slag is

$$P_{stored} = \rho_s V_s C_s \dot{T}_s \tag{2.3}$$

where,

ρ_s : density of the slag (g/cm^3)

V_s : volume of the slag (cm^3)

C_s : specific heat of the slag (J/gK)

We assumed that there is a complete convective mixing inside the molten slag, so fluid motion, electromagnetic, and temperature gradients within the slag are not considered. Therefore, the slag temperature is assumed to be uniform throughout. Equation (2.4) is the state equation describing the heat transfer in terms of the state variable, the slag temperature, where *P*$_{in}$ is input power which is AC and the potential across the slag ranges from 10 to 50 V with a current range from 2,000 to 50,000 amps.

$$\dot{T}_s = \frac{1}{\rho_s \upsilon_s C_s}\begin{bmatrix} P_{in} - H_e A_e \left(T_s - T_m\right) - H_s 2\pi r_m h_s \left(T_s - T_{ss}\right) \\ -\sigma\varepsilon\left(2\pi r_m^2 - 2\pi r_e^2\right)\left(T_s^4 - T_\infty^4\right) \end{bmatrix} \tag{2.4}$$

2.3 MODELING OF THE MELTING DYNAMICS OF THE ELECTRODE

2.3.1 Thermal Energy Model of the Electrode

The modeling of the electrode melting dynamics in ESR share much in common with the modeling of the electrode melting dynamics in VAR. The ESR electrode melting dynamics can be described as a moving boundary Stefan problem similar to the VAR process [8]. Due to these similarities in the processes, the modeling approach used for the VAR process can be used without much modification. The modeling of the electrode melting dynamics is mainly based on Beaman's reduced order modeling of the VAR process [5]. A one-dimensional, transient, and nonlinear heat flow model is being used to simulate the heat transfer of the electrode model assuming that the electrode is considered as a semi-infinite slab. Assume that the tip of the electrode which is immersed in the slag is initially at the melting temperature and area of the tip is uniform during the process (flat electrode assumption). The system can be modeled as a one-dimensional Fourier's equation. See Figure 2.2 for the energy balance and coordinate setup in the electrode.

$$\rho_e(T)C_e(T)\frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(K_e(T)\frac{\partial T}{\partial x}\right) \tag{2.5}$$

where,

ρ_e: density of the electrode material (g/cm^3)

C_e: specific heat of the electrode material (J/gK)

K_e: thermal conductivity of the electrode material (W/cmK)

The boundary conditions for the differential equations are

$$T(S_e,t)=T_m \tag{2.6}$$

$$T(\infty,t)=T_r \tag{2.7}$$

$$\frac{\partial T}{\partial x}(\infty,t)=0 \tag{2.8}$$

$$P_m(I,d)+K_eA_e\frac{\partial T}{\partial x}=\rho_{\text{sup}}L^*A_e\dot{S}_e \tag{2.9}$$

where,

S_e : the melt front where the phase change occurs (cm)

T_m : melt temperature of the electrode (K)

T_r : room temperature (K)

P_m : melt power ($P_{\text{conduction}}$ to electrode) (W)

I : Input current (A)

d : Immersion depth (cm)

ρ_{sup} : the density of the electrode at the superheat temperature (g/cm^3)

L^* : $L^*=h_{\text{sup}}-h_m$, latent heat of the electrode (J/g)

h_{sup} : superheat temperature specific enthalpy of the electrode (J/cm^3)

h_m : melt temperature specific enthalpy of the electrode (J/cm^3)

To solve this one-dimensional, transient, and nonlinear melting problem, numerical methods (finite element methods, finite difference methods, etc.) or approximate integral methods can be used. Numerical methods are difficult to use in control design due to the large number of the coupled equations, so the approximate integral method will be a better choice for the solution technique to get an appropriate model for control design [5].

Stefan problems with moving boundaries are very important in applications such as welding, ablation, and melting. Before numerical approaches were widespread, approximate integral methods were extensively used. A particular problem that is very similar to the ESR melting process is the ablation of heat shields on spacecraft. In this problem, aerofrictional heating on re-entry causes a heat shield to ablate, which gives a moving boundary condition at the shield surface [5].

After using the boundary layer approximation and approximate integral method, the state equations for melting problem can be described as (For the details of this derivation, see Appendix A).

$$\dot{\Delta} = \frac{\alpha_r C_{\Delta\Delta}}{\Delta} - \frac{C_{\Delta p}}{h_m} p_m \tag{2.10}$$

$$\dot{S}_e = -\frac{\alpha_r C_{s\Delta}}{\Delta} + \frac{C_{sp}}{h_m} p_m \tag{2.11}$$

where, Δ is boundary layer thickness (cm), S_e is the melt front where the phase change occurs (cm), α_r is room temperature thermal diffusion coefficient (cm^2/s), h_m is volume specific enthalpy at melt temperature and dimensionless constants are material specific and derived as

$$C_{\Delta\Delta} = \frac{224(\Lambda^* + 1)}{3\Lambda^* + 11}\left(\frac{1}{2} + \frac{\beta h_m}{3}\right) \tag{2.12}$$

$$C_{\Delta p} = \frac{32\Lambda^*}{3\Lambda^* + 11} \tag{2.13}$$

$$C_{s\Delta} = \frac{56\Lambda^*}{3\Lambda^* + 11}\left(\frac{1}{2} + \frac{\beta h_m}{3}\right) \tag{2.14}$$

$$C_{sp} = \frac{11\Lambda^*}{3\Lambda^* + 11} \tag{2.15}$$

where, Λ^* is the Stefan number, β is a diffusivity parameter, p_m is melting power flux (W/cm^2).

$$\Lambda^* = \frac{h_m}{h_{\sup} - h_m} \tag{2.16}$$

$$\beta = \frac{\alpha_m - \alpha_r}{\alpha_r} \tag{2.17}$$

2.3.2 Buoyancy and Immersion Depth Model

The immersion depth is the distance that the electrode protrudes into the slag as illustrated in Figure 2.3.

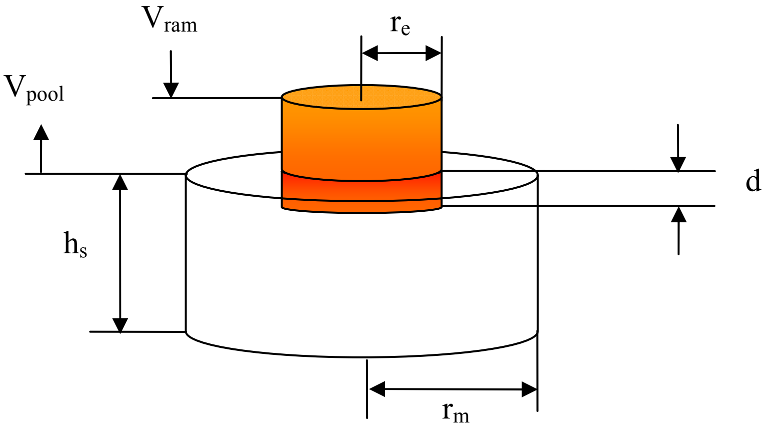


Figure 2.3 Description of the buoyancy and immersion depth model of the ESR process

The volume of the slag at any time is V_s where A_s is slag or ingot diameter, h_s is slag height, and d is immersion depth.

$$V_s = A_s h_s - A_e d \tag{2.18}$$

The volume of the slag is changing very slowly so it is assumed that the volume of the slag remains relatively constant, so the change rate of the slag height is geometrically related to the time derivative of the immersion depth. Here we assumed that the tip of the electrode is flat.

$$\dot{h}_s \approx \frac{A_e}{A_s}\dot{d} \tag{2.19}$$

The melt pool velocity can be defined by the fact that the burn off rate is directly proportional to the pool velocity, V_{pool} (cm/s).

$$V_{pool} = \frac{A_e}{A_s}\dot{S}_e \tag{2.20}$$

and the melt rate is

$$\dot{m}_e = \rho_e A_e \dot{S}_e \tag{2.21}$$

By using the above relationship, the immersion depth can be defined as a function of the burn off rate and the ram velocity.

$$\dot{h}_s - \dot{d} = \dot{S}_e - V_{ram} - V_{pool} \tag{2.22}$$

and finally,

$$\dot{d} = -\dot{S}_e + \frac{1}{a}V_{ram} \tag{2.23}$$

where a, geometry fill ratio factor is,

$$a = 1 - \frac{A_e}{A_s} \tag{2.24}$$

2.4 SIMULATION AND EXPERIMENTATION FOR VERIFICATION OF THE MODEL

The simulation of ESR model was performed on MATLAB using the *ode45* differential equation solver. The state equations and the initial conditions were defined in a MATLAB program and the state variables were predicted over the time period of a melt. Using the simulation and model, experiments were designed to evaluate the physics and the accuracy of the model. The experiment named E139A was performed on an ESR laboratory furnace at Sandia National Laboratories in Albuquerque, New Mexico. The melt lasted approximately one and a half hours and was performed in the spring of 1999. The current control system was used to measure the melt rate, load cell position, and the current input into the ESR process. The current control system measures the values of the melt rate and averages them in five minute intervals. After the averages for every five minute are calculated, a least squares fit is used between data points. This results in a five minute lag in measuring the melt rate data currently and a lag before it is downloaded. AISI 316 steel was used and the experiment was performed in a mold that was ten inches in diameter and thirty inches in height. Cooling water was passed from the top of the mold through to the bottom through the crucible, and the change in water temperature was measured. It was approximately 2-3 K during the length of the melt, which is negligible compared to the change in temperature of the slag (1700-2200K) when melting from room temperature (300K). The experimental data was collected at the sampling rate of 15 Hz. The measured values include: load cell position, input current, velocity of the ram, time, temperature of water at the inlet, temperature of water at outlet, and the power. The noise of each of the measurement instrument was characterized before the melt began. The experimental results were adjusted for the five minute lag time when they were compared to the simulation values. The model verification for the melt rate with the E139A experiment is shown at Figure 2.4.

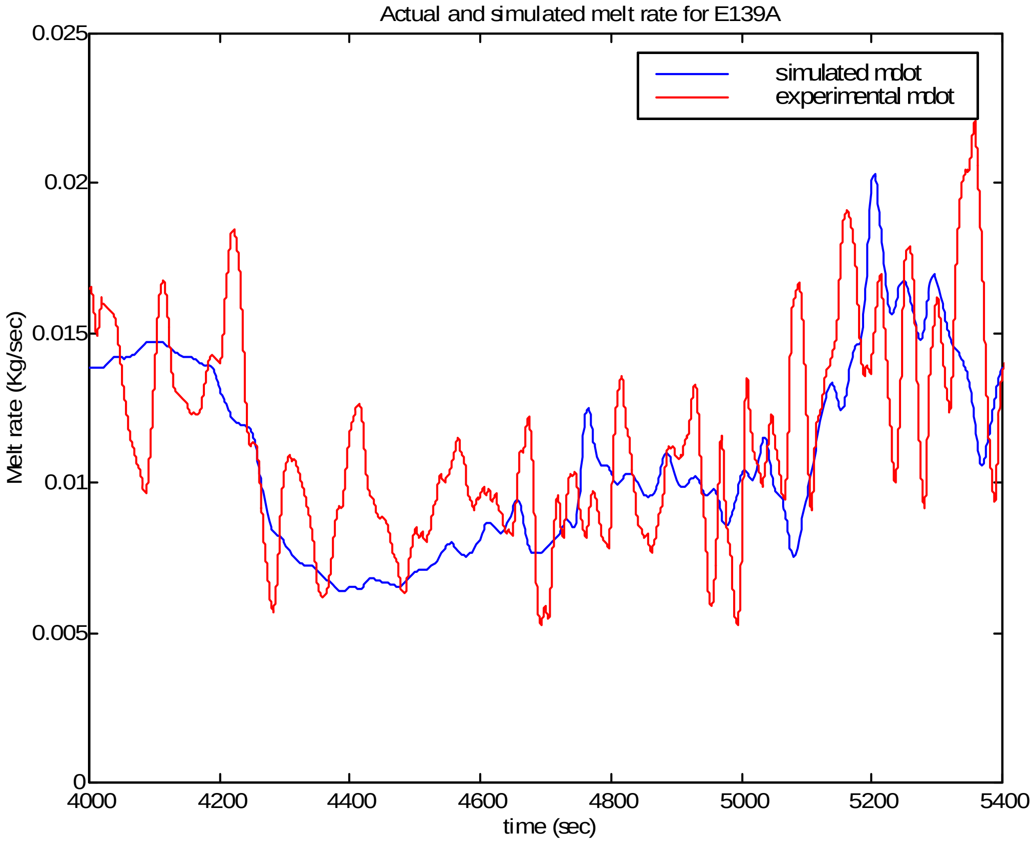


Figure 2.4 Verification of the model using the experimental data E139A

CHAPTER 3

RECURSIVE BAYESIAN ESTIMATION

3.1 BAYESIAN ESTIMATION APPROACH

The purpose of nonlinear estimation is to estimate the evolving "state" of the dynamic system in a recursive fashion when incomplete and noisy measurements are available on-line. We focus on using a state space approach for modeling the evolution of the dynamics system, and we adopt a discrete-time evaluation of the problem. For dynamic state estimation, the discrete-time approach is widely accepted and used for its convenience.

To have a probabilistic inference about the given dynamics system, we need at least two models: A model which describes the evolution of the system with time and a model which relates the noisy measurements to the states. Most of the time, these models are available in a probabilistic form. Let's define $\mathbf{x}_k$ as the state of the system at discrete time index k then $\mathbf{X}_k$, $\mathbf{X}_k = \{\mathbf{x}_i, i = 1,...,k\}$, is a sequence of state variables. $\mathbf{X}_k$ is generally regarded as a random variable due to either noise in the state evolution or the uncertainty of the process itself. We assume that information concerning this unknown state sequence is conveyed through a measurement process. Defining $\mathbf{z}_k$ as the measurement produced by the system at discrete time index k similar to our state notation, let $\mathbf{Z}_k$, $\mathbf{Z}_k = \{\mathbf{z}_i, i = 1,...,k\}$, denote a sequence of measurements. $\mathbf{Z}_k$ is also regarded as a random variable due to either noise in the measurement or uncertainty in the underlying state sequence. For the purpose of monitoring or controlling a given stochastic system, we wish to estimate its evolving state $\mathbf{x}_k$ using all measurements $\mathbf{Z}_k$ collected up to the current time.

Updating the information based on receipt of the recent measurements is well suited for the Bayesian approach. In the framework of the Bayesian approach, a filtering problem is any technique that produces an estimate of the posterior probability density $p(\mathbf{x}_k|\mathbf{Z}_k)$ for each k=1,2,… where $\mathbf{Z}_k = \{\mathbf{z}_i, i = 1,...,k\}$. Because $\mathbf{x}_k$ is a random variable, all available information provided by $\mathbf{Z}_k$ is carried by the posterior probability density. Since this probability density function (pdf) embodies all available statistical information, it may be regarded as the complete solution to the estimation problem. For example, a minimum mean-square error (MMSE) estimate of the current state can be found by computing the conditional mean

$$\hat{\mathbf{x}}_k = \int \mathbf{x}_k p(\mathbf{x}_k|\mathbf{Z}_k) d\mathbf{x}_k \tag{3.1}$$

For many problems, an estimate is required every time a measurement is available. In this case a recursive algorithm has to be implemented. A recursive Bayesian approach means that received data can be processed sequentially rather than as a batch. A recursive Bayesian filtering algorithm imposes the constraint that the estimate of $p(\mathbf{x}_k|\mathbf{Z}_k)$ should be generated only from the previous posterior probability density $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$ and the most recent measurement $\mathbf{z}_k$. In this way, the problem of storing $\mathbf{Z}_k$, the entire measurement sequence, is avoided.

3.2 RECURSIVE BAYESIAN ESTIMATION

In general, it is not possible to perform Bayesian filtering recursively. In order to perform Bayesian filtering recursively, we need to impose constraints on our stochastic systems.

The first assumption we make is that the state sequence is a first order Markov process,

$$p(\mathbf{x}_k | \mathbf{X}_{k-1}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}) \tag{3.2}$$

For Markov 1st order processes, the value of $\mathbf{x}_{k-1}$ provides adequate information about $\mathbf{x}_k$ because it provides information from all of the previous time instants. The second assumption is the observation or measurement information is conditionally independent of the rest of the state sequence and all other measurements. This can be expressed as

$$p(\mathbf{Z}_k | \mathbf{X}_k) = \prod_{i=1}^{k} p(\mathbf{z}_i | \mathbf{x}_i) \tag{3.3}$$

With these two assumptions, the posterior density $p(\mathbf{x}_k | \mathbf{Z}_k)$ can be obtained recursively in the form of two stages: prediction and update. The posterior at discrete time index k-1, $p(\mathbf{x}_{k-1} | \mathbf{Z}_{k-1})$, is projected toward the process model in order to calculate the *prior* or *prediction density* at time k via the Chapman-Komolgorov Equation:

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})d\mathbf{x}_{k-1} \tag{3.4}$$

The probabilistic model of the state evolution $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is defined by the system dynamics model and the known noise statistics.

At time step k when a measurement $\mathbf{z}_k$ becomes available, the update stage is carried out. This involves an update of the prior via Bayes rule and use of the conditional independence of measurement assumption:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{Z}_k) &= \frac{p(\mathbf{Z}_k|\mathbf{x}_k)p(\mathbf{x}_k)}{p(\mathbf{Z}_k)} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{Z}_{k-1}|\mathbf{x}_k)p(\mathbf{x}_k)}{p(\mathbf{Z}_k)} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})p(\mathbf{Z}_{k-1})p(\mathbf{x}_k)}{p(\mathbf{x}_k)p(\mathbf{Z}_k)} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})p(\mathbf{Z}_{k-1})}{p(\mathbf{Z}_k)} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})p(\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})p(\mathbf{Z}_{k-1})} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})} \end{aligned} \tag{3.5}$$

where the normalizing constant

$$p(\mathbf{z}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})d\mathbf{x}_k \tag{3.6}$$

depends on the likelihood function, $p(\mathbf{z}_k|\mathbf{x}_k)$, defined by the measurement model and the known statistics of measurement noises. In the equation (3.5), the measurement $\mathbf{z}_k$ is used to modify the prior density to obtain the required posterior density of the current state.

The recurrence relations (3.4) and (3.5) form the basis of the recursive Bayesian estimation algorithm. In summary, while Bayesian filtering cannot be performed recursively in general, it can be accomplished for stochastic systems with the Markov first order state sequences and conditionally independent measurement process.

Now the recursive Bayesian filtering for the dynamic system can be defined by considering a stochastic system model, formulated in discrete time

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \tag{3.7}$$

where $\mathbf{x}_k$ is an n_x dimensional state vector, and $\mathbf{u}_k$ is an n_u external control input vector, and $\mathbf{w}_k$ is n_w dimensional process noise vector. This is a powerful formulation that allows us to approximate a differential equation of arbitrary order as a first-order vector difference equation. Next, consider a stochastic measurement equation of the form

$$\mathbf{z}_k = h_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k) \tag{3.8}$$

where $\mathbf{z}_k$ is n_z observation vector corresponding to state $\mathbf{x}_k$, $\mathbf{u}_k$ is an external input vector and $\mathbf{v}_k$ is n_v dimensional measurement noise. By adopting a dynamic system model, the assumptions that the $\mathbf{x}_k$ state sequence is a Markov first order sequence and the conditional independency of the measurement information can be restated as

- The process noise vector $\mathbf{w}_k$ forms an independent sequence.
- The measurement noise vector $\mathbf{v}_k$ forms an independent sequence.
- The sequence $\mathbf{W}_k$ and $\mathbf{V}_k$ and the random variable $\mathbf{x}_k$ are mutually independent.

The recursive propagation of the posterior density given by (3.4) and (3.5) is only a conceptual solution in the sense that in general it cannot be determined analytically. The multi-dimensional integrals in equations (3.4), (3.5), and (3.6) are usually only tractable for linear, Gaussian systems only, in which case the closed-form recursive solution is given by the well known Kalman filter. For most general real-world problems (nonlinear, non-Gaussian), we need approximate solutions [39].

3.3 OPTIMAL RECURSIVE BAYESIAN ESTIMATION ALGORITHMS

We showed that if the state sequence is a Markov first order and the measurement information is conditionally independent given the state sequence, then a conceptual solution to the recursive Bayesian filtering problem could be found. But in general, the solution does not enable an engineering solution. However there are several optimal finite-dimensional algorithms for recursive Bayesian state estimation that can be formulated in the following cases:

1. If the state $\mathbf{x}_k$ is drawn from a state space that is both discrete and finite, the integrals in (3.4) and (3.5) become finite sums, which are easily computed.
2. If state equation $f_k(.)$ is linear and Gaussian, the functional recursion of (3.4) and (3.5) becomes the Kalman filter and it provides the exact recursive solution for $p(\mathbf{x}_k|\mathbf{Z}_k)$.
3. For a certain class of nonlinear problems, suggested by Beněs and Daum, it is also possible to formulate exact analytic solutions. (see [7, 10])

3.3.1 The Kalman Filter

For the dynamic system, if the restated assumptions at Section 3.2 are satisfied and the system model takes the linear form

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{w}_k \tag{3.9}$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{3.10}$$

where $\mathbf{F}_k$ and $\mathbf{H}_k$ are appropriately dimensioned matrices. For the sake of simplicity, control inputs are not considered but can be easily added, and we will show it later. $\mathbf{F}_k$ and $\mathbf{H}_k$ represent a dynamic system where the functions are still permitted to vary with time, but are now linear. With this system we can show the Kalman filter is the exact and optimal solution for the recursive Bayesian estimation problem. For the above linear system, if $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$ is Gaussian and the process noise $\mathbf{w}_k$ and the measurement noise $\mathbf{v}_k$ are Gaussian with covariance $\mathbf{Q}_k$ and $\mathbf{R}_k$ then $p(\mathbf{x}_k|\mathbf{Z}_k)$ is also Gaussian. The Kalman filter, derived using equation (3.9) and (3.10), can then be viewed as the following relationship:

$$p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1}) = \mathcal{N}\left(\mathbf{x}_{k-1};\hat{\mathbf{x}}_{k-1},\mathbf{P}_{k-1}\right) \tag{3.11}$$

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \mathcal{N}\left(\mathbf{x}_k;\hat{\mathbf{x}}_k^-,\mathbf{P}_k^-\right) \tag{3.12}$$

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \mathcal{N}\left(\mathbf{x}_k;\hat{\mathbf{x}}_k,\mathbf{P}_k\right) \tag{3.13}$$

where $\mathcal{N}\left(\mathbf{x};\boldsymbol{\mu},\mathbf{P}\right)$ is a Gaussian density with argument **x**, mean **μ,** and covariance **P**.

We can now state the Kalman filter equations:

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_k\hat{\mathbf{x}}_{k-1} \tag{3.14}$$

$$\mathbf{P}_k^- = \mathbf{F}_k\mathbf{P}_{k-1}\mathbf{F}_k^T + \mathbf{Q}_k \tag{3.15}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \tag{3.16}$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \tag{3.17}$$

where

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \tag{3.18}$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{3.19}$$

$\mathbf{S}_k$ is the covariance of the innovation term and $\mathbf{K}_k$ is the Kalman gain. Based on the fact that Gaussian distributions are preserved under linear transformations and a Gaussian distribution is completely specified by mean and the variance, we see the Kalman filtering accomplishes propagation of the conditional mean and the conditional variance by recursive prediction and update under the Bayesian framework [22, 14].

3.3.2 Grid-based approach

If the state space is discrete and consists of a finite number of the states, a grid-based approach provide the optimal recursion of the posterior density, $p(\mathbf{x}_k | \mathbf{Z}_k)$ because integrals from the derivation of the recursive Bayesian estimation become finite sum, which are easily can be computed. Suppose the state space at time k-1 consists of discrete states $\mathbf{x}_{k-1}^i$, $i = 1,\ldots,N$. For each state $\mathbf{x}_{k-1}^i$, let the conditional probability of that state, given measurements up to time k-1, be denoted by w_{k-1}^i that is, $\Pr\left\{\mathbf{x}_{k-1} = \mathbf{x}_{k-1}^i \middle| \mathbf{Z}_{k-1}\right\} \triangleq w_{k-1}^i$. Then the posterior probability density function at k-1 can be written as

$$p(\mathbf{x}_{k-1} | \mathbf{Z}_{k-1}) = \sum_{i=1}^{N} w_{k-1}^i \delta\left(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^i\right) \tag{3.20}$$

where $\delta(.)$ is the Dirac delta function. If we substitute (3.20) into the formulation of the recursive Bayesian estimation derived at the section 3.2, we can have the prediction and update equations in finite sum form

$$p(\mathbf{x}_k \middle| \mathbf{Z}_{k-1}) = \sum_{i=1}^{N} w_k^{i-} \delta\left(\mathbf{x}_k - \mathbf{x}_k^i\right) \tag{3.21}$$

$$p(\mathbf{x}_k \middle| \mathbf{Z}_k) = \sum_{i=1}^{N} w_k^{i} \delta\left(\mathbf{x}_k - \mathbf{x}_k^i\right) \tag{3.22}$$

where

$$w_k^{i-} \triangleq \sum_{j=1}^{N} w_{k-1}^{j} p(\mathbf{x}_k^{\mathrm{i}} \middle| \mathbf{x}_{k-1}^{\mathrm{j}}) \tag{3.23}$$

$$w_k^{i-} \triangleq \frac{w_{k-1}^{j-} p(\mathbf{z}_k \middle| \mathbf{x}_k^{\mathrm{i}})}{\sum_{j=1}^{N} w_{k-1}^{j-} p(\mathbf{z}_k \middle| \mathbf{x}_k^{\mathrm{j}})} \tag{3.24}$$

and where the normalization term in the recursion equation (3.23) requires only the ability to evaluate the transition probabilities $p(\mathbf{x}_k^{\mathrm{i}} | \mathbf{x}_{k-1}^{\mathrm{j}})$ and the likelihood function $p(\mathbf{z}_k | \mathbf{x}_k^{\mathrm{i}})$. Furthermore, the likelihood only needs to be evaluated up to a normalization constant because of its presence in both the numerator and denominator. Grid-based approach also called as exact hidden Markov model (HMM) filter and being widely used in speech recognition algorithms where the state space dynamics are reasonably discrete [38].

3.4 SUBOPTIMAL RECURSIVE BAYESIAN ESTIMATION ALGORITHMS

Unfortunately, most system models in the process control or target tracking applications are continuous, nonlinear, and non-Gaussian problems, and it is typically impossible to implement the recursive Bayesian solution as shown at the previous section. After the Kalman filter was published, a huge number of approximate nonlinear or non-Gaussian filters were suggested. It will be hard to review all of them. Instead, in this section we will review the filters that have been adopted and used widely in scientific and engineering fields. We can categorize them into four subclasses based on the how the filter approximates the posterior probability distribution density.

1. Approximates the posterior as Gaussian distribution. (e.g. extended Kalman filter, unscented Kalman filter)
2. Approximates the posterior as a sum of basis function (e.g. Gaussian sum filter).
3. Approximates the posterior numerically. (e.g. hidden Markov model filter. We already discussed the exact version of this filter as a grid-based approach)
4. Sampling approaches. (e.g. particle filter).

At the end of this chapter, we will discuss how the advantages and disadvantages of each filter can be used with the process control problem to enhance the accuracy of the estimation of the states of the system dynamics.

3.4.1 Extended Kalman Filter

The extended Kalman filter (EKF) is a combination of the Kalman filter algorithm with the 1st order Taylor series approximation technique to deal with the nonlinearities in the system and measurement dynamics model. The dynamic system most often considered is

$$\mathbf{x}_k = f_k\left(\mathbf{x}_{k-1}\right) + \mathbf{w}_k \tag{3.25}$$

$$\mathbf{z}_k = h_k(\mathbf{x}_k) + \mathbf{v}_k \tag{3.26}$$

where $f_k(.)$ and $h_k(.)$ can be nonlinear and $\mathbf{w}_k$ and $\mathbf{v}_k$ are mutually independent, zero-mean white Gaussian random variables with covariance $\mathbf{Q}_k$ and $\mathbf{R}_k$. The nonlinear functions in (3.25) and (3.26) are approximated by the 1st order Taylor series expansions as

$$f_k(\mathbf{x}_{k-1}) = f_k(\hat{\mathbf{x}}_{k-1}) + \tilde{\mathbf{F}}_k(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \text{higher order terms} \tag{3.27}$$

$$h_k(\mathbf{x}_k) = h_k(\hat{\mathbf{x}}_k^-) + \tilde{\mathbf{H}}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \text{higher order terms} \tag{3.28}$$

where $\tilde{\mathbf{F}}_k$ and $\tilde{\mathbf{H}}_k$ are the Jacobian matrices,

$$\tilde{\mathbf{F}}_k \triangleq \left.\frac{\partial f_k(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}} \tag{3.29}$$

$$\tilde{\mathbf{H}}_k \triangleq \left.\frac{\partial h_k(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} \tag{3.30}$$

By using the linear approximation from (3.27) and (3.28), the mean and the covariance of the underlying Gaussian density are computed as follows:

$$\hat{\mathbf{x}}_k^- = f_k(\hat{\mathbf{x}}_{k-1}) \tag{3.31}$$

$$\mathbf{P}_k^- = \tilde{\mathbf{F}}_k \mathbf{P}_{k-1} \tilde{\mathbf{F}}_k^T + \mathbf{Q}_k \tag{3.32}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{z}_k - h_k(\hat{\mathbf{x}}_k^-)\right) \tag{3.33}$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T \tag{3.34}$$

where

$$\mathbf{S}_k = \tilde{\mathbf{H}}_k \mathbf{P}_k^- \tilde{\mathbf{H}}_k^T + \mathbf{R}_k \tag{3.35}$$

$$\mathbf{K}_k = \mathbf{P}_k^- \tilde{\mathbf{H}}_k^T \mathbf{S}_k^{-1} \tag{3.36}$$

Except where the $\mathbf{H}_k$ has been replaced by $\tilde{\mathbf{H}}_k$, the calculation of $\mathbf{S}_k$ and $\mathbf{K}_k$ is identical to the Kalman filtering algorithm. When the approximation by the 1st order Taylor series expansion is not enough, additional terms can be preserved. The Gaussian second order filter, truncated second order Gaussian filter and lots of variants have been suggested [3, 31]. Additional complexities of the algorithm when dealing with higher-order terms prevented their widespread use.

3.4.2 Unscented Kalman Filter

In 1996, Julier and Uhlmann introduced a novel nonlinear estimation algorithm called the Unscented Kalman filtering algorithm (UKF) [25]. The UKF addresses the approximation issues of the EKF. The state distribution is Gaussian based random variables, but is now specified using a minimal set of deterministically chosen sampling points (sigma points). These sampling points completely capture the true mean and covariance of the prior Gaussian random variables, and when propagated through the true nonlinear system, capture the posterior mean and covariance accurately up to the second order for any nonlinearity without the requirement of the derivatives. To explain this filter how works, we will start by revisiting the Kalman filtering algorithm.

3.4.2.1 Kalman Filter Revisit

We derived the Kalman filter algorithm in the previous section as an application of the Bayesian recursive estimation algorithm with the assumptions that all the estimates have independent, Gaussian-distributed errors and the system is linear. However, Kalman's original derivation did not apply Bayes' rule and does not require the operation of any specific error distribution information other than mean and covariance [27].

We will derive the Kalman filtering algorithm again based on Kalman's original idea. Suppose that the estimate at time step *k*-1 is described by the mean $\hat{\mathbf{x}}_{k-1}$ and covariance $\mathbf{P}_{k-1}$. Kalman's original assumptions are that the system random variables could be consistently estimated with minimum variance by recursively updating the first and second moment of the posterior density and the specific form of the estimator is in a linear form [27, 28].

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{W}_k \mathbf{v}_k \tag{3.37}$$

$$\mathbf{v}_k = \mathbf{z}_k - \hat{\mathbf{z}}_k \tag{3.38}$$

where $\mathbf{v}_k$ is called the innovation.

Here we only assume that the mean and the covariance are maintained. The densities are not required to be Gaussian instead we only keep the Gaussian components of these densities in the estimator. For Gaussian distributions, it will be the complete solution. The Kalman filter consists of prediction and update steps. In the prediction step, the filter propagates the estimate from a previous time step *k*-1 to the current time step *k*. The prediction of the state, covariance and the observations are given by

$$\hat{\mathbf{x}}_k^- = E\left[\mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k\right)\right] \tag{3.39}$$

$$\mathbf{P}_k^- = E\left[(\mathbf{x} - \hat{\mathbf{x}}_k^-)(\mathbf{x} - \hat{\mathbf{x}}_k^-)^T\right] \tag{3.40}$$

$$\hat{\mathbf{z}}_k = E\left[\mathbf{h}\left(\hat{\mathbf{x}}_k^-, \mathbf{v}_k\right)\right] \tag{3.41}$$

Let's define the estimation error $\tilde{\mathbf{x}}_k$ as

$$\tilde{\mathbf{x}}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k \tag{3.42}$$

If we substitute the equation (3.37) into (3.42) then

$$\tilde{\mathbf{x}}_k = \tilde{\mathbf{x}}_k^- - \mathbf{W}_k\mathbf{v}_k \tag{3.43}$$

where $\tilde{\mathbf{x}}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^-$.

The updated covariance of this error can be calculated as

$$\begin{aligned}\mathbf{P}_k &= E\left[\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k^T\right] = E\left[\left(\tilde{\mathbf{x}}_k^- - \mathbf{W}_k\mathbf{v}_k\right)\left(\tilde{\mathbf{x}}_k^- - \mathbf{W}_k\mathbf{v}_k\right)^T\right] \\ &= E\left[\tilde{\mathbf{x}}_k^-\tilde{\mathbf{x}}_k^{-T}\right] - E\left[\tilde{\mathbf{x}}_k^-\mathbf{v}_k^T\right]\mathbf{W}_k^T - \mathbf{W}_kE\left[\mathbf{v}_k\tilde{\mathbf{x}}_k^{-T}\right] + \mathbf{W}_kE\left[\mathbf{v}_k\mathbf{v}_k^T\right]\mathbf{W}_k^T \\ &= \mathbf{P}_k^- - \mathbf{P}_k^{xz}\mathbf{W}_k^T - \mathbf{W}_k\mathbf{P}_k^{zx} + \mathbf{W}_k\mathbf{P}_k^{zz}\mathbf{W}_k^T\end{aligned} \tag{3.44}$$

where $\mathbf{P}_k^{xz} = E\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^-\right)\left(\mathbf{z}_k - \hat{\mathbf{z}}_k\right)^T\right]$ is the cross covariance between the state and observation. $\mathbf{P}_k^{zz} = E\left[\left(\mathbf{z}_k - \hat{\mathbf{z}}_k\right)\left(\mathbf{z}_k - \hat{\mathbf{z}}_k\right)^T\right]$ is the covariance of $\mathbf{v}_k$.

The gain matrix $\mathbf{W}_k$ which minimize the error variance can be calculated by minimizing the trace of $\mathbf{P}_k$. This can be done by taking partial derivative of $\mathbf{P}_k$ with respect to the gain matrix $\mathbf{W}_k$ to zero. Using some linear algebra identities, the gain matrix $\mathbf{W}_k$ can be derived as

$$\mathbf{W}_k = \mathbf{P}_k^{xz} {\mathbf{P}_k^{zz}}^{-1} \tag{3.45}$$

Using this gain, the updated covariance (3.44) is

$$\begin{aligned} \mathbf{P}_k &= \mathbf{P}_k^{-} - \mathbf{W}_k \mathbf{P}_k^{zz} \mathbf{W}_k^T \\ &= \mathbf{P}_k^{-} - \mathbf{W}_k {\mathbf{P}_k^{xz}}^T \end{aligned} \tag{3.46}$$

Therefore, the Kalman filter update equations can be applied if several sets of expectations can be calculated. These are the predicted states and covariances $(\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-)$, the predicted observation and its covariance $(\hat{\mathbf{z}}_k, \mathbf{P}_k^{zz})$ and the cross covariance between the prediction and the observation, $\mathbf{P}_k^{xz}$. When all of the system equations are linear, direct substitution into the above equations gives the Kalman filter. When the system is nonlinear, methods for approximating these quantities must be used. Therefore, the problem of applying the Kalman filter to a nonlinear system becomes one of applying nonlinear transformation to mean and covariance estimates.

3.4.2.2 The Unscented Transformation

The unscented transform (UT) is a novel way of calculating the statistics of the random variable which undergoes the nonlinear transformation. It is based on the intuition that *it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation* [25]. A set of points (sigma points) are chosen deterministically so that their mean and the covariance are $\overline{\mathbf{x}}$ and $\mathbf{P}_x$. The nonlinear function is applied to each point, in turn, to yield a set of transformed points. The statistics of the transformed points can then be calculated to form an estimate of the nonlinearly transformed mean and covariance. Let's consider n dimensional random variable vector $\mathbf{x}$ through an arbitrary nonlinear function,

$$\mathbf{y} = g(\mathbf{x}) \tag{3.47}$$

$$\mathbf{y} = g(\overline{\mathbf{x}} + \Delta\mathbf{x}) \tag{3.48}$$

Assume that $\mathbf{x}$ has mean $\overline{\mathbf{x}}$ and covariance $\mathbf{P}_x$ and $\Delta\mathbf{x}$ is a zero-mean random variable with the same covariance $\mathbf{P}_x$. To calculate the mean and the covariance of $\mathbf{y}$ analytically, we first expand the function by using Taylor series expansion around $\overline{\mathbf{x}}$. (for the scalar case)

$$\begin{aligned} & g(\overline{x} + \Delta x) \\ & = g(\overline{x}) + \left.\frac{dg}{dx}\right|_{x=\overline{x}} \Delta x + \frac{1}{2!}\left.\frac{d^2 g}{dx^2}\right|_{x=\overline{x}} \Delta x^2 + \frac{1}{3!}\left.\frac{d^3 g}{dx^3}\right|_{x=\overline{x}} \Delta x^3 + \frac{1}{4!}\left.\frac{d^4 g}{dx^4}\right|_{x=\overline{x}} \Delta x^4 + \ldots \end{aligned} \tag{3.49}$$

after the transformation, the mean value can be determined by applying the expectation operator for each term of the Taylor series. If we assume that the distribution of $\Delta\mathbf{x}$ is symmetrical, all odd ordered moments in this distributions are zero and the result is

$$
\begin{aligned}
\bar{y} &= E[y] \\
&= E\left[g(\bar{x}+\Delta x)\right] \\
&= g(\bar{x}) + \frac{1}{2!}\frac{d^2 g}{dx^2}\bigg| P_x + \frac{1}{4!}\frac{d^4 g}{dx^4}\bigg| E[\Delta x^4] + \ldots
\end{aligned}
\tag{3.50}
$$

However, the linearization used in EKF algorithm truncates this series at the first order and predicts the mean as

$$\bar{y}_{lin} = g(\bar{x}) \tag{3.51}$$

Similarly, the variance of the transformed random variable can be determined by taking expectations of the Taylor series [41].

$$
\begin{aligned}
P_y &= E[(y-\bar{y})^2] \\
&= \left(\frac{dg}{dx}\right)^2 P_x - \left(\frac{1}{2!}\frac{d^2 g}{dx^2}\right)^2 P_x^2 \\
&\quad + E\left[\frac{1}{3!}\frac{dg}{dx}\frac{d^3 g}{dx^3}\Delta x^4 + \frac{1}{2\cdot 2!}\frac{d^2 g}{dx^2}\frac{d^2 g}{dx^2}\Delta x^4 + \frac{1}{3!}\frac{d^3 g}{dx^3}\frac{dg}{dx}\Delta x^4\right] + \ldots
\end{aligned}
\tag{3.52}
$$

The linearization used in EKF predicts the variance as

$$P_y^{lin} = \left(\frac{dg}{dx}\right)^2 P_x \tag{3.53}$$

To calculate the statistics of the $\mathbf{y}$ by using the unscented transformation, first we choose a set of sigma points $\mathcal{S} = \left\{ i = 0,1,\ldots,p : \mathbf{x}^{(i)}, W^{(i)} \right\}$ consisting of

$n+1$ vectors and their associated weights. The weights $W^{(i)}$ can be negative but to provide an unbiased estimate, they obey the condition

$$\sum_{i=0}^{n} W^{(i)} = 1 \tag{3.54}$$

First, the function is applied to each point to yield the set of transformed sigma points

$$\mathbf{y}^{(i)} = g(\mathbf{x}^{(i)}) \tag{3.55}$$

Then the mean of the $\mathbf{y}$ is given by the weighted average of the transformed sigma points

$$\overline{\mathbf{y}} \approx \sum_{i=0}^{n} W^{(i)} \mathbf{y}^{(i)} \tag{3.56}$$

The covariance of the $\mathbf{y}$ is also given by the weighted outer product of the transformed sigma points

$$\mathbf{P}_y \approx \sum_{i=0}^{n} W^{(i)} \left\{\mathbf{y}^{(i)} - \overline{\mathbf{y}}\right\} \left\{\mathbf{y}^{(i)} - \overline{\mathbf{y}}\right\}^T \tag{3.57}$$

Other statistical values after transformation can also be calculated in a similar way. One set of points which satisfies the above conditions consists of $2n_x + 1$ points which satisfies the following conditions

$$\mathbf{x}^{(0)} = \overline{\mathbf{x}} \tag{3.58}$$

$$\mathbf{x}^{(i)} = \overline{\mathbf{x}} + \left(\sqrt{\left(n_x + \kappa\right)\mathbf{P}_{\mathbf{x}}}\right)_i, \quad \text{where } i = 1, ..., n_x \tag{3.59}$$

$$\mathbf{x}^{(i)} = \overline{\mathbf{x}} - \left(\sqrt{\left(n_x + \kappa\right)\mathbf{P}_{\mathbf{x}}}\right)_i, \quad \text{where } i = n_x + 1, ..., 2n_x \tag{3.60}$$

and associated weights are

$$\begin{aligned} W^{(0)} &= \frac{\kappa}{n_x + \kappa} \\ W^{(i)} &= \frac{1}{2(n_x + \kappa)} \end{aligned} \tag{3.61}$$

where n_x is dimension of the random variable $\mathbf{x}$. κ is a scaling parameter and $\left(\sqrt{\left(n_x + \kappa\right)\mathbf{P}_{\mathbf{x}}}\right)_i$ is the *i*th column or row of the matrix square root of the weighted covariance matrix, $\left(n_x + \kappa\right)\mathbf{P}_{\mathbf{x}}$.

From the equation (3.55) the appropriate set of points can be extracted and a Taylor series expansion is applied as follows: (for the scalar case)

$$x^{(0)} = g(\overline{x}) \tag{3.62}$$

$$
\begin{aligned}
x^{(i)} &= g\left(\bar{x}+\left(\sqrt{(n_x+\kappa)\mathbf{P_x}}\right)_i\right) \\
&= g(\bar{x})+\left.\frac{dg}{dx}\right|_{x=\bar{x}}\left(\sqrt{(n_x+\kappa)\mathbf{P_x}}\right)_i+\frac{1}{2!}\left.\frac{d^2g}{dx^2}\right|_{x=\bar{x}}\left(\sqrt{(n_x+\kappa)\mathbf{P_x}}\right)_i^2 \\
&+\frac{1}{3!}\left.\frac{d^3g}{dx^3}\right|_{x=\bar{x}}\left(\sqrt{(n_x+\kappa)\mathbf{P_x}}\right)_i^3+\frac{1}{4!}\left.\frac{d^4g}{dx^4}\right|_{x=\bar{x}}\left(\sqrt{(n_x+\kappa)\mathbf{P_x}}\right)_i^4+\ldots
\end{aligned}
\tag{3.63}
$$

According to the weighted sum of equation (3.56) and the definition of the weights on equation (3.61), the mean for a scalar transformation becomes

$$
\bar{y}_{UT}=g(\bar{x})+\frac{1}{2!}\left.\frac{d^2g}{dx^2}\right|_{x=\bar{x}}P_x+\frac{1}{4!}\left.\frac{d^4g}{dx^4}\right|_{x=\bar{x}}\left(n_x+\kappa\right)P_x^{\,2}+\ldots
\tag{3.64}
$$

which is an exact representation of the mean $\bar{y}$ up to the second order as shown in equation (3.50). Similar results can be obtained by evaluating the covariance.

$$
\begin{aligned}
P_y^{UT} &= \left(\frac{dg}{dx}\right)^2 P_x-\left(\frac{1}{2!}\frac{d^2g}{dx^2}\right)^2 P_x^2 \\
&+\frac{1}{2(n_x+\kappa)}\sum_{i=1}^{2n}\left[\begin{array}{l}\frac{1}{3!}\frac{dg}{dx}\frac{d^3g}{dx^3}E\left[\Delta x^4\right]_i \\ +\frac{1}{2\cdot 2!}\frac{d^2g}{dx^2}\frac{d^2g}{dx^2}E\left[\Delta x^4\right]_i+\frac{1}{3!}\frac{d^3g}{dx^3}\frac{dg}{dx}E\left[\Delta x^4\right]_i\end{array}\right]+\ldots
\end{aligned}
\tag{3.65}
$$

It is shown in [25] that the covariance approximation agrees with the true covariance up to the second order terms in the Taylor series. The detailed derivation for multi dimensional Taylor series expansion and its application to the unscented transformation can be found on Julier's paper [25]. The properties of this algorithm have been studied in detail in several papers [24, 46]. Here we summarize the results from the studies.

1. The matrix square root of a positive-definite matrix is a not unique, any ortho-normal rotation of the sigma point set is a valid set.
2. With a larger set of sigma points, higher order moment information such as skew or kurtosis can be captured but results in increased computational time and resources [24, 52, 41].
3. The estimates of the mean and covariance are exact up to the second order of the Taylor series expansion of $g(\mathbf{x})$ for any nonlinear function as we shown above.
4. Errors are introduced in the third and the higher moments but are scaled by the choice of κ.
5. Cholesky factorization method is usually used to calculate matrix square root in equation (3.59) and (3.60).
6. In the modified version of the unscented transform, the scaled unscented transform has a scaling parameter which can prevent the resulting covariance from becoming non-positive semi-definite [24].

3.4.2.3 The Scaled Unscented Transformation

The scaled unscented transformation (SUT) is a method that scales an arbitrary sigma point set but ensures that the mean and covariance are maintained correctly. It replaces the original set of sigma point set with a transformed set given by

$$\mathbf{x}_{SUT}^{(i)} = \mathbf{x}^{(0)} + \alpha\left(\mathbf{x}^{(i)} - \mathbf{x}^{(0)}\right), \quad i = 0, ..., 2n_x \tag{3.66}$$

where α is a positive scaling parameter which gives an extra degree of freedom to control the scaling the sigma points to guarantee a positive semi-definite covariance. By applying the UT to an auxiliary random variable $\mathbf{z}$ which is related to $\mathbf{x}$ through the nonlinear equation

$$\mathbf{z} = g^{'}(\mathbf{x}, \bar{\mathbf{x}}, \alpha) = \frac{g\left(\bar{\mathbf{x}} + \alpha(\mathbf{x} - \bar{\mathbf{x}})\right) - g(\bar{\mathbf{x}})}{\alpha^2} + g(\bar{\mathbf{x}}) \tag{3.67}$$

Then taking a Taylor series expansion of the mean and covariance of $g^{'}(\mathbf{x}, \bar{\mathbf{x}}, \alpha)$ about $\bar{\mathbf{x}}$ shows that both exactly match up to the second order, with higher terms scaling geometrically with a common ratio of α [24]. It means we have an additional controllable scaling parameter and possible tune up for higher order accuracy. The SUT transformed sigma point set $\mathcal{S}' = \left\{i = 0, 1, ..., p : \mathbf{x}_{SUT}^{(i)}, W_{SUT}^{(i)}\right\}$ and its weights which satisfies the above conditions consists of $2n_x + 1$ points,

$$\mathbf{x}_{SUT}^{(i)} = \mathbf{x}_{SUT}^{(0)} + \alpha\left(\mathbf{x}_{SUT}^{(i)} - \mathbf{x}_{SUT}^{(0)}\right) \tag{3.68}$$

$$W_{SUT}^{(i)} = \frac{W_{SUT}^{(0)}}{\alpha^2} + \left(1 - \frac{1}{\alpha^2}\right), \quad i = 0 \tag{3.69}$$

$$W_{SUT}^{(i)} = \frac{W_{SUT}^{(i)}}{\alpha^2}, \quad i = 1,...,2n_x \tag{3.70}$$

We can combine the sigma point selection scheme and the scaling method together into a single step by setting

$$\lambda = \alpha^2 \left(n_x + \kappa\right) - n_x \tag{3.71}$$

and selecting the sigma points set by

$$\mathbf{x}^{(0)} = \overline{\mathbf{x}} \tag{3.72}$$

$$\mathbf{x}^{(i)} = \overline{\mathbf{x}} + \left(\sqrt{\left(n_x + \lambda\right)\mathbf{P}_{\mathbf{x}}}\right)_i, \quad \text{where } i = 1,...,n_x \tag{3.73}$$

$$\mathbf{x}^{(i)} = \overline{\mathbf{x}} - \left(\sqrt{\left(n_x + \lambda\right)\mathbf{P}_{\mathbf{x}}}\right)_i, \quad \text{where } i = n_x + 1,...,2n_x \tag{3.74}$$

where appropriate weights for the above selection are

$$W_m^{(0)} = \frac{\lambda}{n_x + \lambda}, \quad W_c^{(0)} = \frac{\lambda}{n_x + \lambda} + \left(1 - \alpha^2 + \beta\right), \quad when\, i = 0 \tag{3.75}$$

$$W_m^{(i)} = W_c^{(i)} = \frac{\lambda}{2\left(n_x + \lambda\right)}, \qquad when\, i = 1,...,2n_x \tag{3.76}$$

The weights on the zeroth sigma point directly affect the magnitude of the errors in the higher order (higher than the second order) terms for symmetric prior distribution [24, 46]. So β is introduced which affects the weighting of the zeroth sigma point for the calculation of the covariance. This extra tune up enables the minimization of higher order errors if prior knowledge of the distribution of $\mathbf{x}$ is available. First, we have to choose parameters κ, α and β. A positive value for κ guarantees positive semi-definite covariance. Usually, we can start from $\kappa = 0$ and specific value for κ is not critical [46]. Scaling parameter $\alpha\ (0 \le \alpha \le 1)$ being used to control the size of the sigma point distribution and should be a small number to avoid sampling non-local effects when nonlinearities are strong. β is a non-negative weighting factor which can be used to enhance the accuracy of the higher order moments of the distribution. For a Gaussian prior distribution the optimal value is 2 [24]. The choice of the set of parameters (κ, α, β) is problem specific and empirically can be determined. After setting up the parameters, calculate the sigma points and its weights by using the equations (3.72) to (3.76). Then the sigma points can be propagated through the nonlinear transformation as in equation (3.55). The derived mean and covariance are

$$\overline{\mathbf{y}} \approx \sum_{i=0}^{2n_x} W_m^{(i)} \mathbf{y}^{(i)} \tag{3.77}$$

$$\mathbf{P}_y \approx \sum_{i=0}^{2n_x} W_c^{(i)} \left\{ \mathbf{y}^{(i)} - \overline{\mathbf{y}} \right\} \left\{ \mathbf{y}^{(i)} - \overline{\mathbf{y}} \right\}^T \tag{3.78}$$

Now we apply this deterministic sampling approach to the Kalman filter framework.

3.4.2.4 *Application of the SUT to Recursive Estimation Algorithm: The Unscented Kalman filter*

In Section 3.4.2.1, we reviewed that the Kalman filter basically consists of the following steps:

1. Predict the new state of the system $\hat{\mathbf{x}}_k^-$ and its associated covariance $\mathbf{P}_k^-$ considering the influence of the process noise.
2. Predict the expected observation $\hat{\mathbf{z}}_k$ and the innovation covariance $\mathbf{P}_k^{zz}$ considering the influence of the measurement noise.
3. Predict the cross covariance between the prediction and the observation, $\mathbf{P}_k^{xz}$.

These steps can be formulated by augmenting the state vector and the process and measurement noise models. The augmented vectors are the $n_a = n_x + n_w + n_v$ dimensional vector

$$\mathbf{x}_k^a = \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{w}_k \\ \mathbf{v}_k \end{bmatrix} \tag{3.79}$$

The process and observation models can be rewritten as a function of $\mathbf{x}_k^a$

$$\mathbf{x}_k^a = f^a(\mathbf{x}_{k-1}^a, \mathbf{u}_k) \tag{3.80}$$

$$\mathbf{z}_k = g^a(\mathbf{x}_k^a, \mathbf{u}_k) \tag{3.81}$$

and the SUT uses sigma points that are computed from

$$\hat{\mathbf{x}}_{a,k} = \begin{pmatrix} \hat{\mathbf{x}}_k \\ \mathbf{0}_{n_w \times 1} \\ \mathbf{0}_{n_v \times 1} \end{pmatrix} \tag{3.82}$$

$$\mathbf{P}_{a,k} = \begin{bmatrix} \mathbf{P}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{w,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_{v,k} \end{bmatrix} \tag{3.83}$$

The estimate is correct to the second order and Jacobian calculations are unnecessary. The general formulation of the unscented Kalman filter are now presented

1. The set of sigma points is created by applying a sigma point selection algorithm based on the equation from (3.72)-(3.76) to the augmented system given by (3.79).
2. The transformed set can be calculated by putting each point through the process model

$$\hat{\mathbf{x}}_{a,k}^{(i)} = \mathbf{f}^{a}(\mathbf{x}_{a,k}^{(i)}, \mathbf{u}_k) \tag{3.84}$$

3. The predicted mean is computed as

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2n_x} W_m^{(i)} \hat{\mathbf{x}}_{a,k}^{(i)} \tag{3.85}$$

4. and the predicted covariance is computed as

$$\mathbf{P}_k^- = \sum_{i=0}^{2n_x} W_c^{(i)} \left\{ \hat{\mathbf{x}}_{a,k}^{(i)} - \hat{\mathbf{x}}_k^- \right\} \left\{ \hat{\mathbf{x}}_{a,k}^{(i)} - \hat{\mathbf{x}}_k^- \right\}^T \tag{3.86}$$

5. The prediction of observation can be calculated by putting the transformed set into the measurement model.

$$\hat{\mathbf{z}}_k^{(i)} = \mathbf{h}(\hat{\mathbf{x}}_{a,k}^{(i)}, \mathbf{u}_k) \tag{3.87}$$

6. The predicted observation is calculated by

$$\hat{\mathbf{z}}_k = \sum_{i=0}^{2n_x} W_m^{(i)} \hat{\mathbf{z}}_k^{(i)} \tag{3.88}$$

7. The innovation covariance is

$$\mathbf{P}_k^{zz} = \sum_{i=0}^{2n_x} W_c^{(i)} \left\{ \hat{\mathbf{z}}_k^{(i)} - \hat{\mathbf{z}}_k \right\} \left\{ \hat{\mathbf{z}}_k^{(i)} - \hat{\mathbf{z}}_k \right\}^T \tag{3.89}$$

8. The cross covariance matrix is determined by

$$\mathbf{P}_k^{xz} = \sum_{i=0}^{2n_x} W_c^{(i)} \left\{ \hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k^- \right\} \left\{ \hat{\mathbf{z}}_k^{(i)} - \hat{\mathbf{z}}_k \right\}^T \tag{3.90}$$

9. Finally, the update can be performed using the Kalman filter frame

$$\begin{aligned}
\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{W}_k \mathbf{v}_k \\
\mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{W}_k \mathbf{P}_k^{zz} \mathbf{W}_k^T \\
\mathbf{v}_k &= \mathbf{z}_k - \hat{\mathbf{z}}_k \\
\mathbf{W}_k &= \mathbf{P}_k^{xz} \mathbf{P}_k^{zz^{-1}}
\end{aligned} \tag{3.91}$$

3.4.3 Gaussian Sum Filter

EKF and UKF approximate the posterior density as a Gaussian distribution. For applications such as multi-target tracking, the true posterior is often multimodal. In this case, the Gaussian approximation results in an error by assuming it as a single Gaussian distribution. For multimodal systems, EKF operates more as a maximum likelihood estimator than a minimum variance estimator, because the resulting estimate of the posterior density actually follows just one of the peaks of the true posterior. As an alternative to the single Gaussian approximation, a sum of Gaussian can be used to approximate the true posterior. The Gaussian sum filter uses a weighted sum of Gaussian functions to approximate a posterior density of arbitrary distribution. More details about the Gaussian sum filter can be found in [1]. Application of the Gaussian sum approach for calculating the proposal density function to particle filtering algorithms can be found in [44].

CHAPTER 4

MONTE CARLO BASED APPROACH FOR FILTERING PROBLEMS

We have learned from the previous chapter that the optimal recursive filtering is only available for the two special cases: 1) discrete and finite state space systems or 2) linear, Gaussian systems. In most of the problems encountered in engineering and science, the state space is continuous and the measurement model is nonlinear. We have reviewed the suboptimal algorithms mainly based on the Kalman filtering algorithm, but the variants of the Kalman filter still suffer the drawback of assuming the posterior density as a Gaussian distribution. The Gaussian sum filter can deal with the multimodality, but it uses the Kalman filtering algorithm and has the same drawbacks as the EKF algorithm, especially the 1st order Taylor expansion for approximating nonlinearities.

Particle filters are also suboptimal filtering algorithms. After the seminal paper by Gordon regarding the resampling technique [16], many variants of the particle filters were introduced [12]. It uses sequential Monte Carlo estimation based on a point mass or particle representation of the posterior distribution without making any assumptions to its form thus it can be used with general nonlinear, non-Gaussian systems. Since the speed of computers has been exponentially increasing according to Moore’s law, the use of the filtering algorithms based on the Monte Carlo technique continues to gain popularity. In this chapter we will review the theoretical background of the sequential Monte Carlo technique. The discussion of this chapter mainly draws upon material from [2] and his book [39].

4.1 PERFECT MONTE CARLO INTEGRATION

Let’s begin with the problem of solving a multi-dimensional integral. We want to evaluate the multi-dimensional integral numerically,

$$I = \int g(\mathbf{x}) d\mathbf{x} \tag{4.1}$$

The Monte Carlo methods for numerical integration factorize the function $g(\mathbf{x}) = f(\mathbf{x}) p(\mathbf{x}|\mathbf{z})$ in such a way that $p(\mathbf{x}|\mathbf{z}) \geq 0$ and $\int p(\mathbf{x}|\mathbf{z}) d\mathbf{x} = 1$. Then the integration can be restated as

$$I = \int f(\mathbf{x}) p(\mathbf{x}|\mathbf{z}) d\mathbf{x} \tag{4.2}$$

An approximation, which we denote I_N, can be obtained by Monte Carlo integration,

$$I_N \triangleq \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}^{(i)}) \tag{4.3}$$

where samples $\left\{\mathbf{x}^{(i)}; i=1,...,N\right\}$ are drawn independently from $p(\mathbf{x}|\mathbf{z})$. The validity of (4.3) can be guaranteed by the strong law of large numbers which states that the average of the many independent random variables with common mean and finite variance converges to their common mean almost surely,

$$\lim_{N\to\infty} I_N = I, \text{ with probability one.} \tag{4.4}$$

Moreover, if the variance of $f(\mathbf{x})$, $\sigma^2 = \int \left(f(\mathbf{x}) - I\right)^2 p(\mathbf{x}|\mathbf{z})\, d\mathbf{x}$, is bounded then the central limit theorem holds,

$$\sqrt{N}\left(I_N - I\right) \Rightarrow \mathcal{N}(\mathbf{0}, \sigma^2), \text{ when } N \to \infty \tag{4.5}$$

Equation (4.5) clearly shows how the error in the Monte Carlo approximation is related with the dimension of the state space. The Monte Carlo approximation which samples the state space randomly has an accuracy of $O(N^{-1/2})$ and is independent of the dimension of the state. In contrast, the deterministic numerical integration has an accuracy of $O(N^{-1/n_x})$ and is dependent on the dimension of the state space. It is a very useful and important property of the Monte Carlo approximation because the choice of the samples automatically comes from the state space regions that are important for the integration result. Unfortunately, σ^2 may grow as the dimension of the state space increases. Theoretically Monte Carlo integration is preferred in multi-dimensional spaces rather than deterministic integration but both approximations may be inaccurate depending on how σ^2 scales with the dimension of the state space [11]. Also it is impossible to sample effectively from the posterior density when the posterior density is a non-standard one. We now acknowledge that one of the main tasks of the Monte Carlo approximation is determining how to sample from a target distribution. In the next section, we will discuss the sampling techniques.

4.2 IMPORTANCE SAMPLING ALGORITHM

Suppose we can only generate the samples from a density, $q(\mathbf{x}|\mathbf{z})$, which is similar to $p(\mathbf{x}|\mathbf{z})$ and if $q(\mathbf{x}|\mathbf{z})$ satisfies the condition that

$$q(\mathbf{x}|\mathbf{z}) > 0, \text{ whenever } p(\mathbf{x}|\mathbf{z}) > 0 \qquad (4.6)$$

Then, the integral (4.2) can be written as

$$\begin{aligned} I &= \int f(\mathbf{x}) \frac{p(\mathbf{x}|\mathbf{z})}{q(\mathbf{x}|\mathbf{z})} q(\mathbf{x}|\mathbf{z}) dx \\ &= \frac{1}{p(\mathbf{z})} \int f(\mathbf{x}) \frac{p(\mathbf{z}|\mathbf{x}) p(\mathbf{x})}{q(\mathbf{x}|\mathbf{z})} q(\mathbf{x}|\mathbf{z}) dx \end{aligned} \qquad (4.7)$$

The density $q(\mathbf{x}|\mathbf{z})$ is referred to as the *sampling density* or *proposal density*. Now, a Monte Carlo estimate of I depends on how we draw samples from $\left\{\mathbf{x}^{(i)}; i = 1, \ldots, N\right\}$ according to $q(\mathbf{x}|\mathbf{z})$

$$I_N^* \triangleq \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}^{(i)}) w^*(\mathbf{x}^{(i)}) \qquad (4.8)$$

where

$$w^*(\mathbf{x}^{(i)}) \triangleq \frac{p(\mathbf{x}^{(i)}|\mathbf{z})}{q(\mathbf{x}^{(i)}|\mathbf{z})} \tag{4.9}$$

$w^*(\mathbf{x}^{(i)})$ is referred to as the importance weight. If we apply Bayes' rule to (4.9), the importance weight becomes

$$w^*(\mathbf{x}^{(i)}) = \frac{p(\mathbf{z}|\mathbf{x}^{(i)})p(\mathbf{x}^{(i)})}{p(\mathbf{z})q(\mathbf{x}^{(i)}|\mathbf{z})} \tag{4.10}$$

While it is straightforward to evaluate the prior $p(\mathbf{x}^{(i)})$ and the likelihood $p(\mathbf{z}|\mathbf{x}^{(i)})$, the calculation of the normalization term $p(\mathbf{z})$ is often intractable. In order to avoid the need to evaluate $p(\mathbf{z})$, we use Bayes' rule with equation (4.7), resulting in

$$I = \frac{1}{p(\mathbf{z})}\int f(\mathbf{x})\frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{q(\mathbf{x}|\mathbf{z})}q(\mathbf{x}|\mathbf{z})dx \tag{4.11}$$

$$= \frac{\int f(\mathbf{x})\frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{q(\mathbf{x}|\mathbf{z})}q(\mathbf{x}|\mathbf{z})dx}{\int \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{q(\mathbf{x}|\mathbf{z})}q(\mathbf{x}|\mathbf{z})dx} \tag{4.12}$$

$$= \frac{E_q\left[f(\mathbf{x})\,w(\mathbf{x})\middle|\mathbf{z}\right]}{E_q\left[w(\mathbf{x})\middle|\mathbf{z}\right]} \tag{4.13}$$

where $E_q\left[.\middle|\mathbf{z}\right]$ means the expectation taken with respect to $q(\mathbf{x}|\mathbf{z})$.

$$w(\mathbf{x}) = \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{q(\mathbf{x}|\mathbf{z})} \tag{4.14}$$

Now, the new approximation of I can be evaluated as

$$\hat{I}_N \triangleq \frac{\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^{(i)})w(\mathbf{x}^{(i)})}{\frac{1}{N}\sum_{i=1}^{N} w(\mathbf{x}^{(i)})} \tag{4.15}$$

$$= \sum_{i=1}^{N} f(\mathbf{x}^{(i)})\overline{w}(\mathbf{x}^{(i)}) \tag{4.16}$$

where

$$\overline{w}(\mathbf{x}^{(i)}) \triangleq \frac{w(\mathbf{x}^{(i)})}{\sum_{j=1}^{N} w(\mathbf{x}^{(j)})} \tag{4.17}$$

The estimate of I, $\hat{I}_N$, is biased since it is a ratio of the estimate [39]. However, the strong law of the large numbers still applies and the central limit theorem holds. Thus $\hat{I}_N$ almost surely converges to I as $N \to \infty$. In summary, the importance sampling algorithm is a method to approximate the posterior density without sampling directly from $p(\mathbf{x}|\mathbf{z})$. From this point of view, we can show that the approximation of $p(\mathbf{x}|\mathbf{z})$ can be expressed as

$$\hat{p}(\mathbf{x}|\mathbf{z}) \triangleq \sum_{i=1}^{N} \overline{w}(\mathbf{x}^{(i)})\delta(\mathbf{x}-\mathbf{x}^{(i)}) \tag{4.18}$$

where $\delta(.)$ is the Dirac delta function and the integral I becomes

$$\hat{I}_N = \int f(\mathbf{x})p(\mathbf{x}|\mathbf{z})d\mathbf{x} \approx \int f(\mathbf{x})\hat{p}(\mathbf{x}|\mathbf{z})d\mathbf{x} \tag{4.19}$$

There are other famous sampling techniques such as rejection sampling and Metropolis-Hastings algorithms. For real-time applications, the importance sampling algorithm is the most efficient and does not require a burn-in period which is a weakness of Metropolis-Hastings algorithms [19]. Now we need to extend importance sampling to the sequential version, the sequential importance sampling. Then we can approximate the posterior density sequentially and can adapt the recursive Bayesian estimation algorithm. Ultimately, the goal is to approximate the posterior density at each evolution and calculate the related statistical information in real-time.

4.3 SEQUENTIAL IMPORTANCE SAMPLING ALGORITHM

The sequential importance sampling (SIS) algorithm is a Monte Carlo method that forms the basis for most sequential Monte Carlo filtering algorithms developed over the past decades [12]. This approach is also known as bootstrap filtering [16], the condensation algorithm, particle filtering, interacting particle approximations, and survival of the fittest. It is a technique for implementing recursive Bayesian estimation by using Monte Carlo simulation. The main idea of sequential Monte Carlo filtering algorithm is to represent the wanted posterior density by a set of random particles or samples with associated weights. As the number of the samples increase, this Monte Carlo representation of the posterior density becomes an equivalent description of the true posterior density, and the estimation based on the SIS becomes the optimal Bayesian estimation.

Let's start from the sequence $\mathbf{X}_k = \left\{\mathbf{x}_j, j = 0, ..., k\right\}$ which represents the sequence of all states up to time *k*. The joint posterior density at time *k* is denoted by $p(\mathbf{X}_k|\mathbf{Z}_k)$ and its marginal is $p(\mathbf{x}_k|\mathbf{Z}_k)$. Let $p(\mathbf{X}_k|\mathbf{Z}_k)$ be completely specified by the set $\left\{\mathbf{X}_k^{(i)}, w_k^{(i)}\right\}_{i=1}^{N}$ then

$$p(\mathbf{X}_k|\mathbf{Z}_k) \triangleq \sum_{i=1}^{N} w_k^{(i)}(\mathbf{x}^{(i)})\delta(\mathbf{X}_k - \mathbf{X}_k{}^{(i)}) \tag{4.20}$$

Now, we have a discrete weighted approximation of the true posterior $p(\mathbf{X}_k|\mathbf{Z}_k)$. The normalized weights are derived from the equation (4.14)

$$w_k^{(i)} = \frac{p(\mathbf{Z}_k|\mathbf{X}_k^{(i)})p(\mathbf{X}_k^{(i)})}{q(\mathbf{X}_k^{(i)}|\mathbf{Z}_k)} \tag{4.21}$$

We are trying to approximate $p(\mathbf{X}_k|\mathbf{Z}_k)$ from the previous step $p(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1})$ with the reception of measurement $\mathbf{z}_k$. If the proposal distribution is chosen to factor as

$$q(\mathbf{X}_k|\mathbf{Z}_k) \triangleq q(\mathbf{x}_k|\mathbf{X}_{k-1},\mathbf{Z}_k)q(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1}) \tag{4.22}$$

Substituting equation (4.22) into (4.21) and using the assumptions that the sequence follows the Markov 1st order and conditional independence of the measurement $\mathbf{Z}_k$ results in

$$w_k^{(i)} = \frac{p(\mathbf{Z}_k|\mathbf{X}_k^{(i)})p(\mathbf{X}_k^{(i)})}{q(\mathbf{x}_k^{(i)}|\mathbf{X}_{k-1}^{(i)},\mathbf{Z}_k)q(\mathbf{X}_{k-1}^{(i)}|\mathbf{Z}_{k-1})} \tag{4.23}$$

$$= \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(i)})p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})p(\mathbf{Z}_{k-1}|\mathbf{X}_{k-1}^{(i)})p(\mathbf{X}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)}|\mathbf{X}_{k-1}^{(i)},\mathbf{Z}_k)q(\mathbf{X}_{k-1}^{(i)}|\mathbf{Z}_{k-1})} \tag{4.24}$$

$$= \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(i)})p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)}|\mathbf{X}_{k-1}^{(i)},\mathbf{Z}_k)} w_{k-1}^{(i)} \tag{4.25}$$

$$= \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(i)})p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)},\mathbf{z}_k)} w_{k-1}^{(i)} \tag{4.26}$$

Now the weights can be updated sequentially and SIS algorithm approximates the posterior filtered density $p(\mathbf{x}_k|\mathbf{Z}_k)$ as

$$\hat{p}(\mathbf{x}_k|\mathbf{Z}_k) \triangleq \sum_{i=1}^{N} w_k^{(i)}\delta(\mathbf{x}_k - \mathbf{x}_k{}^{(i)}) \tag{4.27}$$

The SIS algorithm is summarized as a pseudo code in Figure 4.1. The choice of the importance density plays a crucial role in the SIS filtering and it will be discussed in the next section.

```
$\left\{\mathbf{x}_k^{(i)}, w_k^{(i)}\right\}_{i=1}^{N} = \text{SIS}\left[\left\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\right\}_{i=1}^{N}, \mathbf{z}_k\right]$
  • For $i = 1:N$
      o Sample $\mathbf{x}_k^{(i)}$ from $q(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)}, \mathbf{Z}_k)$
      o Evaluate the importance weights $\tilde{w}_k^{(i)}$ based on the
        equation (4.26)
  • End for
  • Calculate the total weights: $w_T = \sum\left[\{w_k^{(i)}\}_{i=1}^{N}\right]$
  • For $i = 1:N$
      o Normalize: $w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{w_T}$
  • End for
```

Figure 4.1 Pseudo code description of the sequential importance sampling

Degeneracy of the SIS algorithm

Conceptually the SIS algorithm can be used with a recursive Bayesian filter but practically it has a serious limitation. Ideally the importance density function should be the posterior distribution itself, $p(\mathbf{x}_k | \mathbf{Z}_k)$. For the importance function of the form (4.22), it has been shown that the variance of the importance weights can only increase over time [12]. It means that after a certain number of recursive steps, all but one particle will have negligible normalized weights. The degeneracy is impossible to avoid in the SIS algorithm and it was a major obstacle for the sequential Monte Carlo methods to overcome [12]. From the computational point of view, it is inefficient to use resources calculating support points and weights $\left\{\mathbf{x}_k^{(i)}, w_k^{(i)}\right\}$ whose contribution to estimates produced by the filter is negligible. A suitable measure of degeneracy of the algorithm is the effective sample size N_{eff} and estimated as follows [29]

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (w_k^{(i)})^2} \tag{4.28}$$

where $w_k^{(i)}$ is the normalized weight obtained by (4.26). If the weights are uniform, $w_k^{(i)} = 1/N$ then $N_{eff} = N$. If $\exists j \in \{1, ..., N\}$ such that $w_k^{(j)} = 1$ and $w_k^{(i)} = 0$ for all $i \neq j$ then $N_{eff} = 1$. Small N_{eff} means a severe degeneracy. To overcome degeneracy issue we will introduce two approaches.

4.4 SELECTION OF THE BETTER PROPOSAL DISTRIBUTION

4.4.1 The Optimal Proposal Distribution

The optimal choice of the proposal distribution is the most critical issue in the design of the particle filter [2]. The optimal proposal distribution that minimizes the variance of importance weights we derived from the last section (4.26) has been shown to be

$$q(\mathbf{x}_k^{(i)} \big| \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)_{optimal} = p(\mathbf{x}_k^{(i)} \big| \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k) \tag{4.29}$$

If we substitute (4.29) into (4.26) then

$$\begin{aligned} w_k^{(i)} &= \frac{p(\mathbf{z}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} \big| \mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)} \big| \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)_{optimal}} w_{k-1}^{(i)} \\ &= \frac{p(\mathbf{z}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} \big| \mathbf{x}_{k-1}^{(i)})}{p(\mathbf{x}_k^{(i)} \big| \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)} w_{k-1}^{(i)} \\ &= \frac{p(\mathbf{z}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} \big| \mathbf{x}_{k-1}^{(i)})}{\dfrac{p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{x}_{k-1}^{(i)}) p(\mathbf{x}_k^{(i)} \big| \mathbf{x}_{k-1}^{(i)})}{p(\mathbf{z}_k \big| \mathbf{x}_{k-1}^{(i)})}} w_{k-1}^{(i)} \\ &= p(\mathbf{z}_k \big| \mathbf{x}_{k-1}^{(i)}) w_{k-1}^{(i)} \end{aligned} \tag{4.30}$$

where we used the conditional independence of the measurement $\mathbf{z}_k$ given $\mathbf{x}_k$. The equation (4.30) states that the weight importance at time *k* can be computed before the particles are propagated to time *k*. There are two problems with this definition of the importance weights. First, it requires sampling from the $p(\mathbf{x}_k^{(i)} \big| \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)$ and it may be a non-standard distribution shape. Second, to evaluate the importance weight $w_k^{(i)}$ it requires the integral to be evaluated

$$p(\mathbf{z}_k \big| \mathbf{x}_{k-1}^{(i)}) = \int p(\mathbf{z}_k \big| \mathbf{x}_k) p(\mathbf{x}_k \big| \mathbf{x}_{k-1}^{(i)}) d\mathbf{x}_k \tag{4.31}$$

This integral may be intractable. However, there are two cases where the use of $q(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)_{optimal}$ is possible. The first case is when $\mathbf{x}_k$ is from a finite state space. The integral becomes a sum and the sampling from $p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)$ is possible. The other case occurs with dynamics with additive Gaussian noise and linear measurement equations. The integral (4.31) can be analytically tractable. Moreover the measurement equation is linear, $p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)$ is also Gaussian and can be practically sampled. Details of the formulation for the two cases can be found in [39].

4.4.2 Suboptimal Choice of the Proposal Distribution

Although it may be far from the optimal; however, a popular choice for the proposal distribution is the *prior* distribution also called as the *transitional prior* [39].

$$q(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k) = p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)}) \tag{4.32}$$

The sampling from the prior is straightforward most of time. For example, if an additive zero-mean Gaussian process noise model is used, the prior is simply

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)}) = \mathcal{N}(\mathbf{x}_k ; f(\mathbf{x}_{k-1}^{(i)}, \mathbf{0}), \mathbf{Q}_{k-1}) \tag{4.33}$$

Substitution of (4.32) into (4.30) yields

$$w_k^{(i)} = p(\mathbf{z}_k | \mathbf{x}_k^{(i)}) w_{k-1}^{(i)} \tag{4.34}$$

In spite of its simplicity and convenience, use of the prior as the proposal distribution can lead to poor performance. If the prior has a much broader distribution than the likelihood $p(\mathbf{z}_k | \mathbf{x}_k)$, then only a few particles will be assigned high weights. As a result, the particles will degenerate rapidly and the filter will not work. Practically, this situation can happen when very accurate and low noise sensors are used [44]. From a different point of view, it means the prior as proposal distribution does not account for the current measurement $\mathbf{z}_k$. Several particle filters using different algorithms to incorporate the current measurement into the proposal have been recently published [37]. It is also possible to use local linearization techniques. Such filtering algorithms use a Gaussian approximation of the proposal distribution based on the extended Kalman filter or the Unscented Kalman filter [44].

4.5 SEQUENTIAL IMPORTANCE RESAMPLING ALGORITHM

As a second approach to overcome the importance sampling degeneracy issues, the resampling algorithm was introduced by Gordon [16]. From the previous section, we have learned how to detect the degeneracy based on the calculation of the effective sample size N_{eff}. Whenever a significant degeneracy is observed that is when N_{eff} falls below a threshold N_{thr}, resampling techniques can be applied to the importance sampling as an additional process. Let's start from the approximate representation of $p(\mathbf{x}_k | \mathbf{Z}_k)$ equation (4.27).

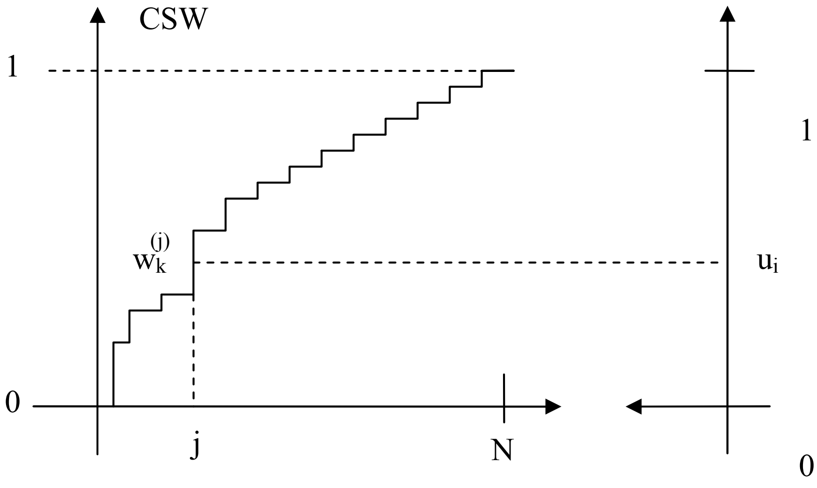
$$\hat{p}(\mathbf{x}_k | \mathbf{Z}_k) \triangleq \sum_{i=1}^{N} w_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}) \tag{4.35}$$

Resampling is a mapping from a random measure $\left\{\mathbf{x}_k^{(i)}, w_k^{(i)}\right\}$ into a random measure $\left\{\tilde{\mathbf{x}}_k^{(i)}, 1/N\right\}$ with uniform weights. The new set of random particles are generated by resampling N times from equation (4.35) such that

$$\Pr(\tilde{\mathbf{x}}_k^{(i)} = \mathbf{x}_k^{(j)}) = w_k^{(j)} \tag{4.36}$$

Because resampling draws from the true posterior (when $N \to \infty$), the resampled importance weights are uniform. Thus, resampling prevents the importance sampling from degenerating by constructing a new random measure where all support points have weight $1/N$. The equation (4.36) means resampling tends to multiply those states with significant weights while discarding those with negligible weights. The selection of $\tilde{\mathbf{x}}_k^{(i)} = \mathbf{x}_k^{(j)}$ is schematically shown in Figure 4.2. CSW stands for cumulative sum of weights of random measure $\left\{\mathbf{x}_k^{(i)}, w_k^{(i)}\right\}$ and random variable $u_i, i = 1, \ldots, N$ is uniformly distributed on the interval [0,1] [44].

In general, the method of resampling consists of generating N independent and identically distributed variables from the uniform distribution sorting by the ascending order and comparing them with the cumulative sum of the normalized weights. There are several different resampling algorithms are introduced in the literature [12]. For our particle filtering implementation, we choose residual resampling algorithm because of its simplicity of implementation and computational efficiency. Also, it has a lower sample variance [44]. In general the specific choice of the resampling scheme does not significantly affect the performance of the particle filter [46].



j= resampled index, i= sampling index

Figure 4.2 A schematic description of resampling algorithm

Although the resampling technique reduces the effects of degeneracy, it introduces two new problems. First, it limits the ability to parallelize the implementation since all the particles must be summed during normalization. Second, the particles with high importance weight are statistically selected many times during the resampling, leading to a loss of diversity. This problem known as *sample impoverishment* is severe in the case where process noise in the dynamics is very small. It leads to the situation where all particles will collapse to a single point within a few iterations. Additional procedures to alleviate this sample depletion issue are needed. A single Markov chain Monte Carlo (MCMC) step can resolve this issue [15]. Details of the MCMC are beyond the scope of this dissertation. If our application of particle filtering to the system shows severe sample depletion issues, MCMC step can be applied.

4.6 THE PARTICLE FILTERING ALGORITHM

Since we have discussed all the background information for the particle filtering algorithm, it is time to organize and review the concepts we have been using. The optimal proposal distribution which minimizes the variance of the importance weights is given by the equation (4.29). Sampling from this optimal proposal distribution is practically impossible because of difficulties in the sampling from the nonstandard distribution and the intractable integral. Instead we can use the prior as the proposal distribution. The effectiveness of this proposal distribution depends on how close the prior is to the true posterior distribution. If there is not much overlap between the prior and the true posterior distribution, only a few particles will have significant importance weights. The pseudo code of the particle filtering algorithm is summarized in Figure 4.3.

```
$\left\{\mathbf{x}_k^{(i)}, w_k^{(i)}\right\}_{i=1}^{N} =$ Particle Filter $\left[\left\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\right\}_{i=1}^{N}, \mathbf{z}_k\right]$
    • For $i = 1 : N$
        o Draw $\tilde{\mathbf{x}}_k^{(i)} \sim q(\mathbf{x}_k^{(i)} \,|\, \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)$
        o Evaluate $w_k^{(i)} = p(\mathbf{z}_k \,|\, \mathbf{x}_{k-1}^{(i)}) w_{k-1}^{(i)}$
    • End for
    • For $i = 1 : N$
        o Evaluate normalized weight $\overline{w}_k^{(i)} \triangleq \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}$
    • End for
    • Calculate $N_{eff} = \frac{1}{\sum_{i=1}^{N} (w_k^{(i)})^2}$
    • If $N_{eff} < N_{thr}$
        o $\left\{\mathbf{x}_k^{(i)}, w_k^{(i)}\right\}_{i=1}^{N} =$ resample $\left[\left\{\tilde{\mathbf{x}}_k^{(i)}, w_k^{(i)}\right\}_{i=1}^{N}\right]$
    • else
        o $\left\{\mathbf{x}_k^{(i)}, w_k^{(i)}\right\}_{j=1}^{N} = \left\{\tilde{\mathbf{x}}_k^{(i)}, w_k^{(i)}\right\}_{i=1}^{N}$
    • End if
```

Figure 4.3 Pseudo code description of the generic particle filtering algorithm

CHAPTER 5

OBSERVER BASED CONTROL DESIGN OF THE ESR PROCESS

5.1 INTRODUCTION

Producing high quality ESR ingots with improved yield requires accurate, transient control of the melt rate and immersion depth. This is difficult because a steady state temperature distribution in the electrode has not been achieved, or it has been driven away from the steady state during the process [6]. In industrial practice, several process variables can be monitored in real-time. These include the input voltage, melting current, impedance, ram position, and the electrode mass. The electrode mass information is used to estimate the melt rate, and modern melt rate controllers use these estimates as feedback to control the melt rate. The electrode mass is measured with a load cell transducer and it is very noisy because of the mechanical stiction, the electromagnetic forces in the system, changing buoyancy forces, and other sources. To address this problem, the data are usually filtered, buffered, and fit using a running nonlinear least square regression. The resulting slope is used as the average melt rate over the analysis time. It is not uncommon to use 10 to 20 minutes of data for this analysis. Thus, although a melt rate estimate may be logged several times a second by the controller, each estimate represents 10 to 20 minutes of melt history, resulting in melt rate estimates that lag the ESR process time by 5 to 10 minutes. A controller using these estimates does not work under highly dynamic process conditions, where it is necessary to control on a shorter time scale or commensurate with the time over which the load cell data are buffered and analyzed. This chapter describes the design of robust dynamic ESR process control capable of implementing dynamic, nonlinear current schedules to account for the rapidly changing thermal conditions of the process under noisy environments.

5.2 CONTROLLER STRUCTURE

The controller is designed with a model-based philosophy. It is based on the premise that physical understanding of the process should be combined with measurements to provide an optimal estimate if the process is to be controlled. Once a good estimate of these quantities is obtained, controlling the system is often straightforward [6].

The melting dynamics of the electrode can be effectively modeled as a Stefan moving boundary problem successfully derived in Chapter 2. The results is a system of first order ordinary differential equations describing the dynamics of the melting electrode in terms of the two state variables, thermal boundary layer thickness and the electrode immersion depth, and two inputs, current and the electrode ram speed.

$$\dot{\Delta} = \frac{\alpha_r C_{\Delta\Delta}}{\Delta} - \frac{C_{\Delta p}}{h_m} p_m \tag{5.1}$$

$$\dot{d} = \frac{C_{s\Delta}\alpha_r}{\Delta} - \frac{C_{sp}}{h_m} p_m + \frac{1}{a} V_{ram} \tag{5.2}$$

where, dimensionless constants are material specific and as follows

$$C_{\Delta\Delta} = \frac{224(\Lambda^* + 1)}{3\Lambda^* + 11}\left(\frac{1}{2} + \frac{\beta h_m}{3}\right) \tag{5.3}$$

$$C_{\Delta p} = \frac{32\Lambda^*}{3\Lambda^* + 11} \tag{5.4}$$

$$C_{s\Delta} = \frac{56\Lambda^*}{3\Lambda^* + 11}\left(\frac{1}{2} + \frac{\beta h_m}{3}\right) \tag{5.5}$$

$$C_{sp} = \frac{11\Lambda^*}{3\Lambda^* + 11} \tag{5.6}$$

where, Λ^* is the Stefan number

$$\Lambda^* = \frac{h_m}{h_{\sup} - h_m} \tag{5.7}$$

and the diffusivity parameter β that relates enthalpy per volume h to diffusivity according to

$$\alpha = \alpha_r (1 + \beta h) \tag{5.8}$$

The slag temperature is also derived in Chapter 2 and neglecting the radiation term will be

$$\dot{T}_s = \frac{1}{\rho_s \upsilon_s C_s}\left[P_{in} - H_e A_e\left(T_s - T_m\right) - H_s 2\pi r_m h_s\left(T_s - T_{ss}\right)\right] \tag{5.9}$$

p_m in equation (5.1) is the melt power flux, which can be determined by

$$p_m = (1+\mu_r)H_e(T_s - T_m) \tag{5.10}$$

where, μ_r is relative melt efficiency, H_e is the heat transfer coefficient from slag to electrode, and T_s is determined from the equation (5.9). P_{in} is composed of the input current and resulting voltage,

$$P_{in} = VI = R(d,\sigma(T_s))I^2 + V_b I \tag{5.11}$$

V_b is the voltage bias term and the resistance $R\left(d,\sigma(T_s)\right)$ can be experimentally determined as (See, Appendix B for the derivation of this relationship)

$$\begin{aligned} R &= (R_1 - m_0 d)e^{\left[-Aelect\left(T_s - T_{s*}\right)\right]},\ d < 0 \\ R &= (R_1 - m_1 d)e^{\left[-Aelect\left(T_s - T_{s*}\right)\right]},\ else \end{aligned} \tag{5.12}$$

where, R_1 is impedance at inflection, m_0 and m_1 are impedance coefficients, A_{elect} is resistivity temperature coefficient, and T_{s*} is nominal slag temperature for the voltage fit. See Appendix B for more details about these coefficients.

Since immersion depth is a state, it can be obtained directly from the solution of the differential equations. Melt rate, on the other hand, is not a state, but it can be expressed as a function of state variables and inputs as

$$\dot{m} = \rho_e A_e \dot{S} = \rho_e A_e \left(-\frac{C_{s\Delta}\alpha_r}{\Delta} + \frac{C_{sp}}{h_m} p_m \right) \tag{5.13}$$

Electrode position also used to characterize the dynamics of the process though it is not necessary because of the ability to estimate electrode immersion depth using equation (5.2). The electrode position is simply described by

$$\dot{X}_{ram} = V_{ram} \tag{5.14}$$

Based on the derived model, an estimator to estimate all the related state variables and a nonlinear control algorithm are developed. We will talk about the estimator design later this chapter. Here, assume that the estimator is good. The essence of the melt rate and immersion depth control problem for a ESR process can be stated as follows:

Given the desired value for the reference outputs melt rate, $\dot{m}_r$ *and immersion depth,* d_r*; determine the time varying current,* I_c*, and ram speed,* $V_{ram,c}$*, commands required to establish and maintain these outputs.*

Conceptually, this can be done using the derived model. The normal causality for these equations is to input current and ram speed as functions of time and then integrate to obtain immersion depth and thermal boundary layer thickness, and thus melt rate. Control requires inverting this normal causality. For these relatively simple equations, this can be done analytically by first solving equation (5.13) for the necessary melt power flux, $p_{m,r}$. The result is

$$p_{m,r} = \frac{C_{sp}}{h_m}\left(\frac{\dot{m}_r}{\rho_e A_e} + \frac{C_{s\Delta}\alpha_r}{\hat{\Delta}}\right) \tag{5.15}$$

where $\hat{\Delta}$ is the estimated boundary layer thickness from the estimator. The desired $p_{m,r}$ is related to the slag thermodynamics. By using the heat transfer coefficient from slag to electrode, the relationship with the melting power flux and the resulting slag temperature can be expressed as

$$T_{s,r} = T_m + \frac{p_{m,r}}{(1+\hat{\mu}_r)H_e} \tag{5.16}$$

where $\hat{\mu}_r$ is estimated relative melting efficiency. It shows how much energy is being used to melt relative to the total energy delivered to the electrode. Adding a correction term to assure T_s converges to $T_{s,r}$ results in the nominal power which assures the temperature converges to $T_{s,r}$

$$P_{in,r} = H_e A_e\left(\hat{T}_s - T_m\right) + H_s 2\pi r_m h_s\left(\hat{T}_s - T_{ss}\right) - \rho_s \upsilon_s C_s K_{T_s}(\hat{T}_s - T_{s,r}) \tag{5.17}$$

where K_{T_s} is a proportional gain which can be determined based on the time response of the slag temperature relative to the steady state value. From equation (5.11) the desired command input current, I_c, can be determined as follows,

$$I_c = I_r - \hat{I}_b = -\frac{\hat{V}_b}{2\hat{R}} + \sqrt{\left(-\frac{\hat{V}_b}{2\hat{R}}\right)^2 + \frac{P_{in,r}}{\hat{R}}} - \hat{I}_b \tag{5.18}$$

where $\hat{R}$ is the estimated resistance and $\hat{I}_b$ is the estimated current bias term. The desired command ram speed $V_{ram,c}$ can be determined when substituting I_r from the above derivation to the model,

$$\hat{V} = \hat{R}I_r + \hat{V}_b \tag{5.19}$$

Then the estimated total power input can be determined as

$$\hat{P} = \hat{V}I_r \tag{5.20}$$

The estimated heat transfer to the electrode, $\hat{Q}_m$, is needed to calculate the estimated melt power flux, $\hat{p}_m$.

$$\hat{Q}_m = H_e A_e (\hat{T}_s - T_m) \tag{5.21}$$

$$\hat{p}_m = (1 + \hat{\mu}_r)\frac{\hat{Q}_m}{A_e} \tag{5.22}$$

Also the estimated heat transfer to the slag can be determined as

$$\hat{Q}_s = H_s 2\pi r_m h_s (\hat{T}_s - T_{s*}) \tag{5.23}$$

where T_{s*} is a nominal slag temperature for the voltage fit. The estimated burn off rate is derived as

$$\hat{\dot{S}} = -\frac{\alpha_r C_{sd}}{\hat{\Delta}} + \frac{C_{sp}}{h_m}\hat{p}_m \tag{5.24}$$

Adding a correction term to assure $\hat{d}$ converges to d_r results in the desired command ram speed which assures the d_r converges is

$$\begin{aligned} V_{ram,c} &= V_{ram,r} - \hat{V}_{ram,b} \\ &= \hat{a}\hat{\dot{S}} - \hat{a}K_d(\hat{d} - d_r) - \hat{V}_{ram,b} \end{aligned} \tag{5.25}$$

where K_d is a proportional gain which can be determined based on the time response of the immersion depth relative to the steady state.

Given the equations (5.18)-(5.25), the ESR process controller can be constructed if accurate estimates of the state variables are available. There are three primary states, d, Δ, T_s. A ram position state, X_{ram}, and a mass of the electrode, M_e, are added in order to use position and load cell information as measurements. In order to account for variations in melting due to disturbances from steady state, the relative melt efficiency, μ_r, is added to the state as well. Specifying relative melt efficiency as a part of the dynamic problem provides a convenient variable for tracking heat conduction disturbances in the electrode. We cannot control this variable during the normal melting condition so it will be treated as a disturbance variable and its time derivative is set to zero. A change in the estimate of the relative melt efficiency is interpreted as a melting disturbance. The area fill ratio, a, is also being considered as a state to account for variations in the electrode geometry and possible defects in the electrode. The electrodes used in the ESR process for the decontamination of radionuclide application consist of several nested pipes, welded concentrically to a top plate [33]. The variation in the contact area in the slag will cause lots of disturbances. By considering these disturbances due to geometrical changes as a disturbance variable, we can design a more robust control system for the ESR process to decontamination application. Finally, to account for the inaccuracies and bias in the current, voltage, and the electrode ram drive inputs, current bias (I_b), voltage bias (V_b), and ram speed bias (V_{ramb}) are added as states. The state vector and control input vector for the ESR process used in the present design of the control and estimator can then be expressed as

$$\mathbf{x} = \begin{Bmatrix} \Delta \\ T_s \\ d \\ X_{ram} \\ M_e \\ \mu_r \\ a \\ V_b \\ I_b \\ V_{ram_b} \end{Bmatrix}, \quad \mathbf{u} = \begin{Bmatrix} I_c \\ V_{ram_c} \end{Bmatrix} \tag{5.26}$$

and the nonlinear dynamics of ESR problem is given by

$$f(\mathbf{x},\mathbf{u}) = \left\{ \begin{array}{c} \dot{\Delta} \\ \dot{T}_s \\ \dot{d} \\ \dot{X}_{ram} \\ \dot{M}_e \\ \dot{\mu}_r \\ \dot{a} \\ \dot{V}_b \\ \dot{I}_b \\ \dot{V}_{ram_b} \end{array} \right\} = \left\{ \begin{array}{c} \dfrac{C_{\Delta\Delta}\alpha_r}{\Delta} - \dfrac{C_{\Delta p}}{h_m} p_{m,r} \\ \dfrac{1}{\rho_s \upsilon_s C_s} \left\{ P_{in,r} - H_e A_e \left(T_s - T_m \right) - H_s 2\pi r_i h_{s0} \left(T_s - T_{ss} \right) \right\} \\ \dfrac{C_{s\Delta}\alpha_r}{\Delta} - \dfrac{C_{sp}}{h_m} p_m + \dfrac{1}{a}(V_{ram,c} + V_{ram,b}) \\ V_{ram,c} + V_{ram,b} \\ \rho A_e \left(\dfrac{C_{sp}}{h_m} p_{m,r} - \dfrac{C_{s\Delta}\alpha_r}{\Delta} \right) \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{array} \right\} \quad (5.27)$$

The proposed controller structure with selected estimator is illustrated in Figure 5.1. where the nonlinear function blocks used for the illustration are

$$f_1 = \frac{T_m + \dfrac{C_{sp}}{h_m} \left(\dfrac{\dot{m}_r}{\rho_e A_e} + \dfrac{C_{s\Delta}\alpha_r}{\hat{\Delta}} \right)}{(1 + \hat{\mu}_r) H_e} \quad (5.28)$$

$$f_2 = H_e A_e \left(\hat{T}_s - T_m \right) + H_s 2\pi r_m h_s \left(\hat{T}_s - T_{ss} \right) \quad (5.29)$$

$$f_3 = -\frac{\hat{V}_b}{2\hat{R}} + \sqrt{\left(-\frac{\hat{V}_b}{2\hat{R}} \right)^2 + \frac{P_{in,r}}{\hat{R}}} \quad (5.30)$$

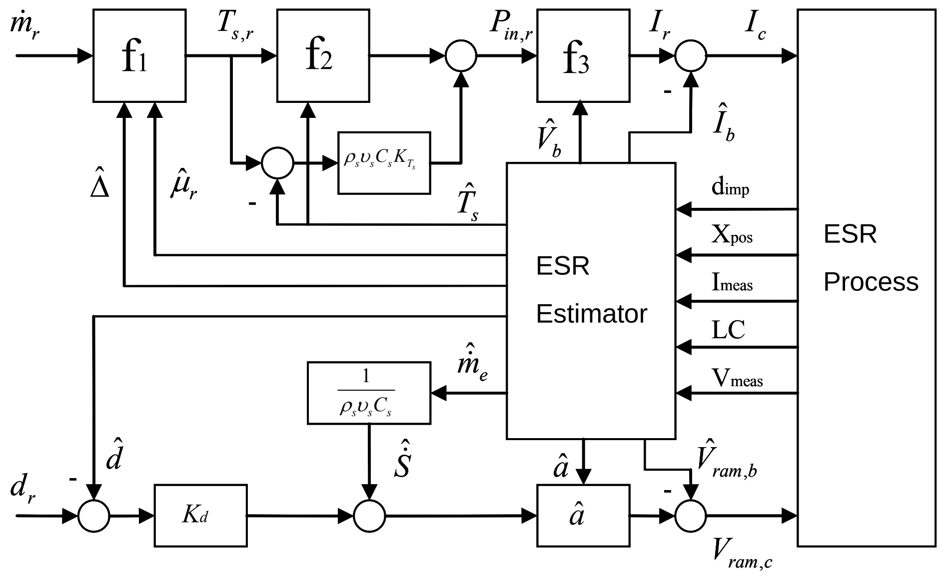


Figure 5.1 Illustration of the control algorithm of ESR process

5.3 ESTIMATOR DESIGN

We suggested four possible candidates for the estimation algorithm from Chapters 3 and 4. First, we will consider a time-invariant Kalman filter. The control algorithm derived in the previous section is based on the fact that the estimates of the state variables are accurate. If it is not then the controller performance will be affected. Therefore, we need an estimator which can track all the nonlinear phenomena happening during the process accurately and fast enough for real-time applications.

Based on the model derived in Chapter 2, we can design an estimator based on the Kalman filtering algorithm. The VAR estimators developed by Beaman et al. [6] and an early adaptation of the estimation algorithm for the ESR application by Beaman and others [4] are based on this theory. The estimators used to control the VAR process can be classified as time-invariant Kalman filter. The Kalman gain is derived off-line from the nominal states and remains constant during implementation. If the dynamics vary significantly from the nominal states, then the possibility of the errors increases. To mitigate this error, nonlinear state propagation has been used but still assumes the Kalman gain is constant [6]. It has shown good results when predicting the gap distance and the melt rate in the VAR processes and further researches are on going. The controller has shown good results in the highly nonlinear period during the process such as a start up and hot topping, but significant opportunities exist for enhancing the controller performance using more sophisticated filtering algorithms.

5.3.1 Steady State Kalman Filter Design

For ESR, using Euler 1st order approximation, the state propagation part of the time-invariant discrete time Kalman filter can be expressed as (More advanced numerical integration technique such as nth order Runge-Kutta methods can be used for this purpose as well, but we found the simpler Euler 1st order approximation to be more than sufficient for the calculation of the ESR dynamics model)

$$\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_{k-1} + f_k\left(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k\right)T_s \tag{5.31}$$

where T_s is a sampling time and the measurement update is

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{z}_k - h_k(\hat{\mathbf{x}}_k^-, \mathbf{u}_k)\right) \tag{5.32}$$

To calculate $\mathbf{K}_k$, we need Jacobian matrices from the dynamics. From Section 3.4.1, we showed how to derive these matrices. For ESR application, the Jacobian matrix $\mathbf{F}_k$ can be determined by linearizing the nonlinear dynamics derived from equation (5.27) for the nominal values. The nominal values being used in the ESR estimation are determined by specifying a nominal immersion depth d_0 and a nominal melt rate $\dot{m}_0$. The appropriate nominal values are organized in Table 5.1.

Table 5.1 Nominal Values

Parameter	Description	Values
μ_0	Nominal melt efficiency	0.55
$\dot{m}_0$	Nominal melt rate	50 (g/sec)
P_{m0}	Nominal melt power	77995.19 (W)
P_0	Nominal total power	141809.43 (W)
p_{m0}	Nominal melt power flux	240.51 (W/cm2)
T_{s0}	Nominal slag temperature	2200 (K)
d_0	Nominal immersion depth	0 (cm)
h_{s0}	Initial slag height	44.24 (cm)
H_e	Heat transfer coefficient from slag to electrode	0.577 (W/cm2K)

Parameter	Description	Values
H_s	Heat transfer coefficient from slag to slag skin	0.042 (W/cm2K)
R_0	Nominal resistance	0.00538 (ohm)
I_0	Nominal current	5121.86 (A)
V_0	Nominal voltage	27.633 (V)
Δ_0	Nominal boundary layer thickness	14.585 (cm)
$\dot{S}_0$	Initial burn off rate	0.021 (cm/sec)
V_{ram0}	Initial ram velocity	0.0075 (cm/sec)
μ_{r0}	Nominal relative efficiency	0

The calculated Jacobians are

$$\mathbf{F}_x = \left.\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_0}$$
$$= \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} & 0 & 0 & 0 & \mathbf{F}_{16} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{F}_{22} & \mathbf{F}_{23} & 0 & 0 & 0 & 0 & \mathbf{F}_{28} & \mathbf{F}_{29} & 0 \\ \mathbf{F}_{31} & \mathbf{F}_{32} & 0 & 0 & 0 & \mathbf{F}_{36} & \mathbf{F}_{37} & 0 & 0 & \mathbf{F}_{310} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{F}_{410} \\ \mathbf{F}_{51} & \mathbf{F}_{52} & 0 & 0 & 0 & \mathbf{F}_{56} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.33)$$

where the subscript $_0$ represents a nominal value and non-zero terms in $\mathbf{F}_x$ are

$$
\begin{aligned}
\mathbf{F}_{11} &= \frac{-\alpha_r C_{\Delta\Delta}}{\Delta_0^2} \\
\mathbf{F}_{12} &= \frac{-C_{\Delta p}}{h_m}\frac{\partial p_m}{\partial T_s} \\
\mathbf{F}_{16} &= \frac{-C_{\Delta p}}{h_m}\frac{\partial p_m}{\partial \mu_r} \\
\mathbf{F}_{22} &= \frac{1}{\rho_s \upsilon_s C_s}\left(\frac{\partial P_{in}}{\partial T_s} - H_e A_e - 2\pi H_s r_i h_{s0}\right) \\
\mathbf{F}_{23} &= \frac{1}{\rho_s \upsilon_s C_s}\frac{\partial P_{in}}{\partial d} \\
\mathbf{F}_{28} &= \frac{1}{\rho_s \upsilon_s C_s}\frac{\partial P_{in}}{\partial V_b} \\
\mathbf{F}_{29} &= \frac{1}{\rho_s \upsilon_s C_s}\frac{\partial P_{in}}{\partial I_b}
\end{aligned}
\tag{5.34}
$$

where

$$\frac{\partial p_m}{\partial T_s} = (1+\mu_{r0})H_e$$

$$\frac{\partial p_m}{\partial \mu_r} = H_e(T_{s0} - T_m)$$

$$\frac{\partial R}{\partial d} = -m_1 e^{\{-A_{elect}(T_{s0}-T_m)\}}$$

$$\frac{\partial R}{\partial T_s} = -(R_1 - m_1 d_0)A_{elect}e^{\{-A_{elect}(T_{s0}-T_m)\}}$$

$$\frac{\partial P_{in}}{\partial T_s} = \frac{\partial R}{\partial T_s} I_0^2$$

$$\frac{\partial P_{in}}{\partial d} = \frac{\partial R}{\partial d} I_0^2$$

$$\frac{\partial P_{in}}{\partial V_b} = I_0$$

$$\frac{\partial P_{in}}{\partial I_b} = R_0 I_0 + V_0 \tag{5.35}$$

$$\mathbf{F}_{31} = \frac{-\alpha_r C_{s\Delta}}{\Delta_0^2}$$

$$\mathbf{F}_{32} = \frac{-C_{sp}}{h_m}\frac{\partial p_m}{\partial T_s}$$

$$\mathbf{F}_{36} = \frac{-C_{sp}}{h_m}\frac{\partial p_m}{\partial \mu_r}$$

$$\mathbf{F}_{37} = -\left(\frac{1}{a_0^2}\right)V_{ram0}$$

$$\mathbf{F}_{310} = \frac{1}{a_0}$$

$$\mathbf{F}_{410} = 1$$

$$\mathbf{F}_{51} = \frac{\rho_m A_e \alpha_r C_{s\Delta}}{\Delta_0^2}$$

$$\mathbf{F}_{52} = \frac{\rho_m A_e C_{sp}}{h_m}\frac{\partial p_m}{\partial T_s}$$

$$\mathbf{F}_{56} = \frac{\rho_m A_e C_{sp}}{h_m}\frac{\partial p_m}{\partial \mu_{rs}} \qquad (5.36)$$

The Jacobian for control input noise matrix is

$$\mathbf{F}_u = \left.\frac{\partial f(\mathbf{x})}{\partial u}\right|_{\mathbf{u}=\bar{\mathbf{u}}}$$
$$= \begin{bmatrix} 0 & 0 \\ \frac{1}{\rho_s \upsilon_s C_s}\left(R_o I_0 + V_0\right) & 0 \\ 0 & \frac{1}{a_0} \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.37}$$

The Jacobian for the process noise input matrix is

$$\mathbf{F}_w = \left.\frac{\partial f(\mathbf{x})}{\partial \mathbf{w}}\right|_{\mathbf{w}=\bar{\mathbf{w}}}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\rho_s \upsilon_s C_s}\left(R_o I_0 + V_0\right) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{a_0} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.38}$$

Five measurements were used for the controller. At present, there is no direct method for determining the actual immersion depth, so it must be inferred from a relationship with the electrical signals. By using the voltage, resistance and the immersion depth relationship (See Appendix B), we can determine the immersion depth alternative, d_{imp} . Similarly for VAR application, the relationship between the melting current, drip-short frequency and gap also used to derive an electrode gap for measurement information [51]. The ram position can be measured by using the encoder attached to the electrode, X_{ram}. Melting current obtained from a Hall effect transducer, I_{meas}. The mass of the electrode measured with load cells, *LC*. For simulation, *LC* will be derived as

$$\begin{aligned} LC &= M_e - \rho_s A_e d, \quad d > 0 \\ &= M_e, \quad else \end{aligned} \tag{5.39}$$

The process voltage can be measured with separate Labview interface with additional voltage divider. The measurement vector **z** is

$$\mathbf{z} = \begin{bmatrix} d_{imp} \\ X_{ram} \\ I_{meas} \\ LC \\ V_{meas} \end{bmatrix} \tag{5.40}$$

The Jacobian for the measurement dynamics, $\mathbf{H}_x$, is derived as (for simulation purpose)

$$\begin{aligned} \mathbf{H}_x &= \left. \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_0} \\ &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\rho_s A_e & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \dfrac{\partial R}{\partial T_s} I_0 & \dfrac{\partial R}{\partial d} I_0 & 0 & 0 & 0 & 0 & 1 & R_0 & 0 \end{bmatrix} \end{aligned} \tag{5.41}$$

If we consider the control noise input, w_{I_c} and $w_{V_{ramc}}$ then,

$$\mathbf{H}_u = \left.\frac{\partial h(\mathbf{x})}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}_0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ R_0 & 0 \end{bmatrix} \tag{5.42}$$

With the derived Jacobian matrices, we can design a steady state Kalman filter for the ESR process. Notice that the Kalman gain $\mathbf{K}_k = \mathbf{K}$ is a constant matrix.

5.3.2 Extended Kalman Filter Design

The extended Kalman filter implementation is very similar to the steady state Kalman filter. Instead of using nominal values for the Jacobians calculation, the EKF evaluates the Jacobian matrices at the current time step based on the estimated states from the previous time step. The details of the equations are already derived from Chapter 3. In summary, the Kalman gain $\mathbf{K}_k$ becomes time-varying gain. The Jacobian matrix $\tilde{\mathbf{F}}_k$ can be determined by linearizing the nonlinear dynamics derived from equation (5.27) for the previous time estimates.

$$\tilde{\mathbf{F}}_k \triangleq \left.\frac{\partial f_k(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}}$$

$$= \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} & 0 & 0 & 0 & \mathbf{F}_{16} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{F}_{22} & \mathbf{F}_{23} & 0 & 0 & 0 & 0 & \mathbf{F}_{28} & \mathbf{F}_{29} & 0 \\ \mathbf{F}_{31} & \mathbf{F}_{32} & 0 & 0 & 0 & \mathbf{F}_{36} & \mathbf{F}_{37} & 0 & 0 & \mathbf{F}_{310} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{F}_{410} \\ \mathbf{F}_{51} & \mathbf{F}_{52} & 0 & 0 & 0 & \mathbf{F}_{56} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.43}$$

where,

$$\begin{aligned} \mathbf{F}_{11} &= \frac{-\alpha_r C_{\Delta\Delta}}{\hat{\Delta}^2} \\ \mathbf{F}_{12} &= \frac{-C_{\Delta p}}{h_m}\frac{\partial \hat{p}_m}{\partial T_s} \\ \mathbf{F}_{16} &= \frac{-C_{\Delta p}}{h_m}\frac{\partial \hat{p}_m}{\partial \mu_r} \\ \mathbf{F}_{22} &= \frac{1}{\rho_s \upsilon_s C_s}\left(\frac{\partial \hat{P}_{in}}{\partial T_s} - H_e A_e - 2\pi H_s r_i h_{s0}\right) \\ \mathbf{F}_{23} &= \frac{1}{\rho_s \upsilon_s C_s}\frac{\partial \hat{P}_{in}}{\partial d} \\ \mathbf{F}_{28} &= \frac{1}{\rho_s \upsilon_s C_s}\frac{\partial \hat{P}_{in}}{\partial V_b} \\ \mathbf{F}_{29} &= \frac{1}{\rho_s \upsilon_s C_s}\frac{\partial \hat{P}_{in}}{\partial I_b} \end{aligned} \tag{5.44}$$

$$
\begin{aligned}
\mathbf{F}_{31} &= \frac{-\alpha_r C_{s\Delta}}{\hat{\Delta}^2} \\
\mathbf{F}_{32} &= \frac{-C_{sp}}{h_m}\frac{\partial \hat{p}_m}{\partial T_s} \\
\mathbf{F}_{36} &= \frac{-C_{sp}}{h_m}\frac{\partial \hat{p}_m}{\partial \mu_r} \\
\mathbf{F}_{37} &= -\left(\frac{1}{\hat{a}^2}\right)V_{ram,r} \\
\mathbf{F}_{310} &= \frac{1}{\hat{a}} \\
\mathbf{F}_{410} &= 1 \\
\mathbf{F}_{51} &= \frac{\rho_m A_e \alpha_r C_{s\Delta}}{\hat{\Delta}^2} \\
\mathbf{F}_{52} &= \frac{\rho_m A_e C_{sp}}{h_m}\frac{\partial \hat{p}_m}{\partial T_s} \\
\mathbf{F}_{56} &= \frac{\rho_m A_e C_{sp}}{h_m}\frac{\partial \hat{p}_m}{\partial \mu_{rs}}
\end{aligned}
\tag{5.45}
$$

and where

$$
\begin{aligned}
&\frac{\partial \hat{p}_m}{\partial T_s} = (1+\hat{\mu}_r)H_e \\
&\frac{\partial \hat{p}_m}{\partial \mu_r} = H_e(\hat{T}_s - T_m) \\
&\frac{\partial \hat{R}}{\partial d} = -m_1 e^{\left\{-A_{elect}(\hat{T}_s - T_m)\right\}} \\
&\frac{\partial \hat{R}}{\partial T_s} = -(R_1 - m_1\hat{d})A_{elect}e^{\left\{-A_{elect}(\hat{T}_s - T_m)\right\}} \\
&\frac{\partial \hat{P}_{in}}{\partial T_s} = \frac{\partial \hat{R}}{\partial T_s} I_r^2 \\
&\frac{\partial \hat{P}_{in}}{\partial d} = \frac{\partial \hat{R}}{\partial d} I_r^2 \\
&\frac{\partial \hat{P}_{in}}{\partial V_b} = I_r \\
&\frac{\partial \hat{P}_{in}}{\partial I_b} = \hat{R}I_r + \hat{V}
\end{aligned}
\tag{5.46}
$$

The Jacobian for the noise input matrix and the measurement are the same as the steady state Kalman filter case, equation (5.41) and (5.42). The Jacobian for the measurement dynamics, $\tilde{\mathbf{H}}_k$, is derived as

$$
\begin{aligned}
\tilde{\mathbf{H}}_k &= \left.\frac{\partial h_k(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} \\
&= \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & -\rho_s A_e & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\partial \hat{R}}{\partial T_s} I_r & \frac{\partial \hat{R}}{\partial d} I_r & 0 & 0 & 0 & 0 & 1 & \hat{R} & 0
\end{bmatrix}
\end{aligned}
\tag{5.47}
$$

5.3.3 The Unscented Kalman Filter Design

One of the benefits using UKF is that UKF does not use any Jacobians. This is beneficial because the calculation of Jacobians may be cumbersome to derive and being a main error source of the implementation. We can directly use the derived nonlinear dynamics equations. For the implementation of the ESR process, the scaled unscented transformation based Unscented Kalman filter has been used. Scaling parameters for the scaled unscented transform, the following values are used for the simulation.

$$\begin{aligned} &\alpha = 1 : \text{Point scaling parameter} \\ &\beta = 2 : \text{Scaling parameter for higher} \\ &\qquad\quad \text{order terms of Taylor series expansion} \\ &\kappa = 0 : \text{Sigma point selection scaling paramter} \end{aligned} \tag{5.48}$$

5.3.4 Particle Filter Design

The sequential importance resampling based on the particle filtering approach has been used for the ESR control design. The prior is being used as a proposal distribution. Programming is based on the pseudo code description from Figure 4.3 in Chapter 4. The residual resampling has been chosen for the resampling technique, the other technique such as systematic resampling [12] and multinomial resampling [46] have been tried, too. We have found that the specific choice of the resampling scheme does not affect the performance of the particle filter.

5.3.5 Process and Measurement Uncertainties

For the design of the estimator, it is necessary to estimate both the model and measurement uncertainties. Experimental data are used to make these estimates. Specifically, for the process inputs, there is unbiased noise on the input current and ram speed. The standard deviation of this noise is experimentally obtained from current and processed ramp position data when commanding constant current and ram speed. The relative efficiency, area ratio, current bias, voltage bias, ram speed bias are treated as random walk processes. For a random walk process, the mean of the variable does not change over time, but its variance increases linearly with time. This means for example, that the change in current bias, dI_b during time step T_s will have a zero mean but a variance that increases linearly in time as shown

$$dI_b = d\beta, E[d\beta] = 0, E[d\beta^2] = \sigma_{I_b}^2 T_s \tag{5.49}$$

In this equation, dI_b is the change in current bias and $d\beta$ is a change in a random walk process (Brownian process) with strength $\sigma_{I_b}^2$. If $T_s = 1$ second, then the current bias would be a normal distribution with nominal mean current I_{b0} and standard deviation σ_{I_b}. Measurement are not perfect and these uncertainties also need to be estimated [6]. The values from the experiments are summarized in Table 5.2.

Table 5.2 Measurement Noise Strength

Parameter	Description	Values
σ_{dV}	Immersion depth standard deviation	0.05 (cm)
σ_{pos}	Encoder standard deviation	0.3 (cm)
$\sigma_{\mathrm{Im}eas}$	Hall effect current sensor standard deviation	200 (A)
σ_{LC}	Load Cell standard deviation	500 (g)
σ_{V}	Ram speed standard deviation	(V)

CHAPTER 6

RESULTS AND DISCUSSION

The suggested controller with four different choices of observers described in Chapter 5 (KF, EKF, UKF, and PF) has been successfully applied to the ESR application. All the parameters used in this chapter are based on a laboratory size ESR furnace. The ingot diameter is 25.4 cm, electrode diameter is 20.32 cm, AISI 1018 steel electrode, and the slag is 60/20/20 slag [Unpublished]. The details of the dimensional parameters and the slag properties are summarized in Table 6.1-6.6. In practice, under normal, steady state conditions, a reasonably constant melt rate is produced by applying constant melting power. However, this is not the case during the start-up and end melting (hot-top). During the initial stages of the ESR process, the temperature distribution in the electrode has not had sufficient time to achieve steady state. As a result, melt rate continually changes under constant power conditions until the steady state temperature distribution is achieved. Toward the end of melting when the melt zone approaches the end of the electrode, heat conduction is impeded because of the large difference in electrode and stub diameters. This causes heat to build up in the electrode end, driving the temperature distribution and therefore, the melt rate, away from steady state. In either case, a nonsteady temperature distribution in the electrode tip leads directly to a nonsteady melt rate at constant power [50].

Table 6.1 Dimensional parameter for electrode

Parameter	Description	Values
D_e	Electrode diameter	20.32 (cm)
D_i	Ingot diameter	25.4 (cm)
A_e	Electrode area	324.29 (cm2)
A_i	Ingot area	506.70 (cm2)

Table 6.2 Electrical parameter

Parameter	Description	Values
A_{elect}	Resistivity temperature coefficient	0.001213 (1/K)
R_1	Impedance at inflection	0.005 (ohm)
m_0	Negative impedance per cm (d<0)	1/70 (ohm/cm)
m_1	Negative impedance per cm (d>0)	1/1300 (ohm/cm)
T_{s*}	Nominal slag temperature for voltage fit	2200 (K)

Table 6.3 Thermophysical Properties of AISI 1018 steel electrode

Parameter	Description	Values
T_r	Room temperature	300 (K)
ρ_r	Room temperature density	7.83 (g/cm3)
C_r	Room temperature specific heat	0.434 (J/gK)
K_r	Room temperature thermal conductivity	0.639 (W/cmK)
α_r	Room temperature diffusivity $\frac{K_r}{\rho_r C_r}$	0.188 (cm2/sec)
T_m	Liquidus temperature	1783 (K)
ρ_m	Melt temperature density	7.40 (g/cm3)
C_m	Melt temperature specific heat	1.168 (J/gK)
K_m	Melt temperature thermal conductivity	0.313 (W/cmK)
α_m	Melt temperature diffusivity $\frac{K_m}{\rho_m C_m}$	0.036 (cm2/sec)
h_m	Approximate melt temperature specific enthalpy, $\frac{1}{2}\left(\rho_m C_m + \rho_r C_r\right)\left(T_m - T_r\right)$	8928.71 (J/cm3)
$T_{\sup}$	Superheat temperature $T_m + 100$	1883 (K)
L	Latent heat	271.96 (J/g)
$h_{\sup}$	Approximate superheat specific enthalpy $\frac{1}{2}\left(\rho_m C_m + \rho_r C_r\right)\left(T_{\sup} - T_r\right) + \rho_m L$	11543.29 (J/cm3)

Table 6.4 Thermophysical Properties of slag

Parameter	Description	Values
C_s	Specific heat of slag	1.47 (J/gK)
ρ_s	Density of slag	2.55 (g/cm3)
K_s	Thermal conductivity of slag	0.0418 (W/cmK)
T_{ss}	Solidus temperature of slag	1773 (K)

Table 6.5 Physical Properties of slag

Parameter	Description	Values
M_s	Mass of slag	56750 (g)
V_s	Volume of slag	22254.90 (cm3)

Table 6.6 Dimensionless parameter and model parameter

Parameter	Description	Values
β_m	Diffusivity parameter	-0.807
Λ	Stefan number	3.415
a_0	Nominal area ratio $1-\frac{A_e}{A_i}$	0.36
den	Model parameter	21.245
$C_{\Delta\Delta}$	Model parameter	10.747
$C_{\Delta P}$	Model parameter	5.144
$C_{S\Delta}$	Model parameter	2.078
C_{SP}	Model parameter	1.768

To investigate the performance, consistency and applicability for real-time implementation of each estimation algorithm in the suggested control design, we consider two different melt profiles. First, we run a linear ramp melt profile to check the controller's performance in transient situations. As subcategories for the suggested melt rate profile, we will investigate two different cases for noise dynamics. From the simulation results, we will choose the candidate estimation algorithm for the ESR process, and it will be applied to a melt profile being used in the industry which includes all three regions (start-up, transient, and hot-top) together. In summary;

1. Run linear ramp melt profile (Constant melt rate then linear ramp up and down back to constant melt rate)
    a. Control input noises (w_{I_c} , $w_{V_{ramc}}$) only

   b. Control input noises (w_{I_c} , $w_{V_{ramc}}$) and other process noises (w_{μ_r} , w_a , w_{V_b} , w_{I_b} , $w_{V_{ram_b}}$)
   c. Particle filter performance comparison compared to the UKF's
2. Choose the estimation algorithm for the ESR control application
3. Run melt profile from Hoyle (includes start-up and hot-top) with Control input noises (w_{I_c} , $w_{V_{ramc}}$) and other process noises (w_{μ_r} , w_a , w_{V_b} , w_{I_b} $w_{V_{ram_b}}$)

By minimizing the noise dynamics, it will be easier to observe the accuracy of each estimation algorithm. For the control input noise only and linear ramp melt profile case (Test 1a), the true values (i.e. true states calculated from the reduced order nonlinear model with uncertainties) will be fed back to the control signal so each estimator's fundamental algorithmic performance can be compared. In other words, the causality reasoning nonlinear controller uses the true known states of the system for the online calculation of the control law. The LKF, EKF, and UKF estimated states were not fed back to the control system. By using this high-fidelity simulation method, it is easy to compare different estimation approaches under different conditions.

To evaluate the performance of the suggested control design as a whole system, the estimates from the suggested estimation algorithm will be used to calculate the control inputs by using the causality reasoning nonlinear controller.

6.1 TEST 1A: LINEAR RAMP MELT PROFILE WITH CONTROL INPUT NOISES ONLY

The discrete version of the dynamics equation (5.31) from Chapter 5 is used to simulate the dynamics of the system. Measurement noise dynamics are also already explained by equation (5.38) from Chapter 5. Assuming the control input noises are the only process noises, the process noise covariance matrix **Q** becomes

$$\mathbf{Q} = \begin{bmatrix} \sigma_{I_c}^2 & 0 \\ 0 & \sigma_{Vram}^2 \end{bmatrix} \tag{6.1}$$

Measurement covariance matrix is given as

$$\mathbf{R} = \begin{bmatrix} \sigma_{dV}{}^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{pos}{}^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{Imeas}{}^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{LC}{}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{V}{}^2 \end{bmatrix} \tag{6.2}$$

Table 6.7 and 6.8 contain the list of the noise strength used in this simulation. The details of the implementation procedure for the simulation algorithm are explained as a Matlab pseudo-code in Appendix C.

Table 6.7 Parameters for Simulation

Parameter	Description	Values
k_0	Impedance coefficient, $\frac{1}{m_0}$	70
k_1	Impedance coefficient, $\frac{1}{m_1}$	1300
R_1	resistance values	0.006
R_{1e}	resistance values	0.0065
$d_{Inflection}$	Immersion depth at inflection	0

Table 6.8 Parameters for Control

Parameter	Description	Values
SlagTempConstant	Time constant for slag temperature	10 (sec)
K_{T_s}	Control gain for Ts	0.1
DepthControlConstant	Time constant for Depth Control	1 (sec)
K_d	Control gain for d	1

The first part of Test 1a is to check the controller performance in transient conditions when the estimates are all perfect. Figure 6.1 shows the melt rate reference used for the test. The melt rate starts from the nominal value at 50 (g/sec) and after 500 seconds ramps up to 70 (g/sec) at a linear rate of 0.5 (g/sec/sec). This melt rate is held for 400 seconds, then ramped back to 50 (g/sec) at a rate of 0.5 (g/sec/sec) until t=1000 seconds and stays at 50 (g/sec) until t=1500 seconds. The immersion depth remains at 0.001 (cm) for the whole simulation period. The result for melt rate control performance is shown in Figure 6.1. After each linear ramp up and down, the melt rate reaches the reference melt rate in 400 seconds. Input current is highly responsive and shows a nonlinear profile.

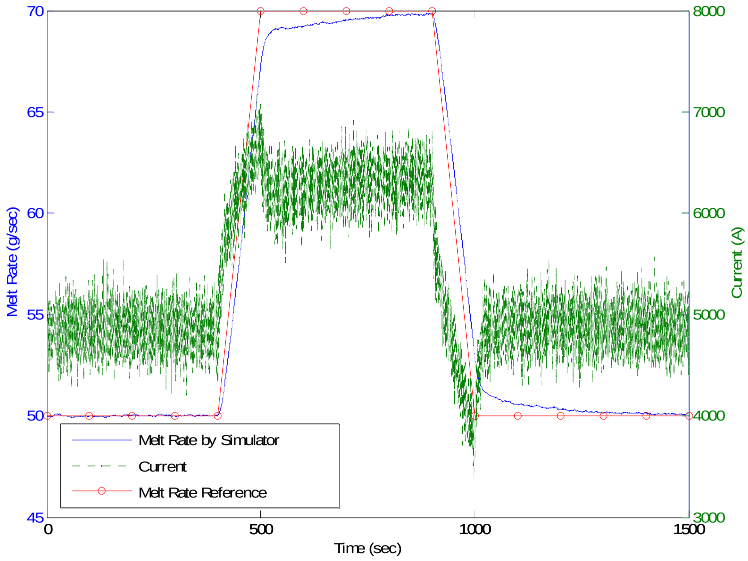


Figure 6.1 Plots of melt rate and input current from Test 1a

Figure 6.2 shows the immersion depth and ram velocity. The immersion depth is noisy but matches with the given reference immersion depth. We will talk about the immersion depth control in detail later this section. The same trend also observed in the ram velocity.

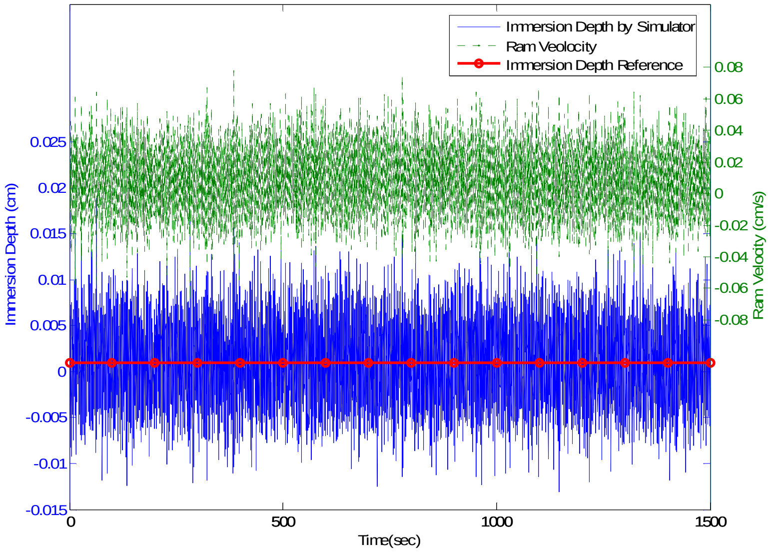


Figure 6.2 Plots of immersion depth and ram velocity of Test 1a

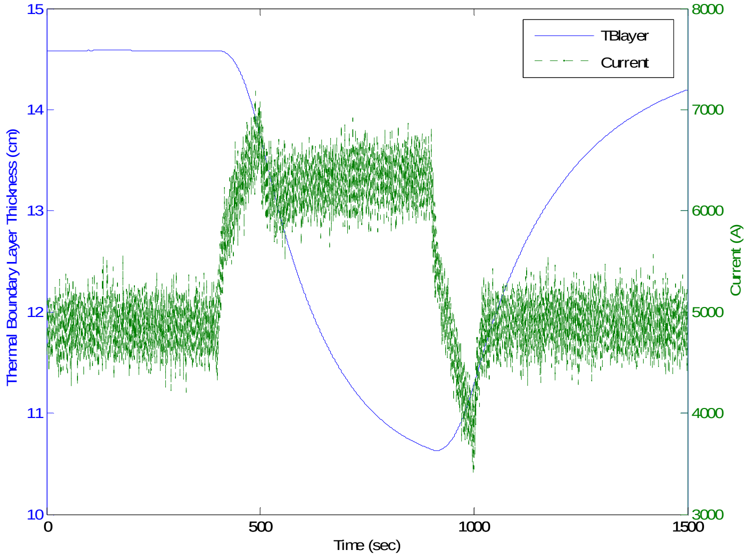


Figure 6.3 Plots of thermal boundary layer thickness and input current of Test 1a

Figure 6.3 shows the thermal boundary layer thickness and input current.

Notice that the thermal boundary layer grows shorter in response to the increased current but hardly reaches steady state after the current ramps back down. In Figure 6.4, slag temperature shows a trend similar to the thermal boundary layer thickness. Also from the load cell data, we can observe the slope change where the linear melt rate ramp has occurred. From the above results, we can conclude that the suggested controller works well if we have accurate estimates.

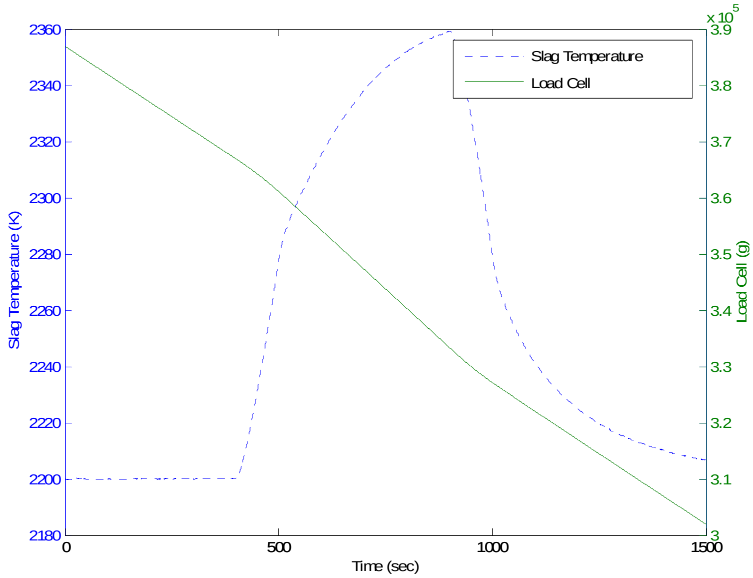


Figure 6.4 Plots of slag temperature and load cell data of Test 1a

Now investigate how well the suggested estimation algorithms perform in Test 1a. We already mentioned that the control inputs are determined by the true states from the simulator dynamics. Figure 6.5 shows the thermal boundary layer thickness estimation results from the different estimation algorithms. Except for the estimates from the UKF, the estimates from the LKF and EKF show some bias, and the estimates from the LKF even diverges. The estimation performance by the UKF is outstanding.

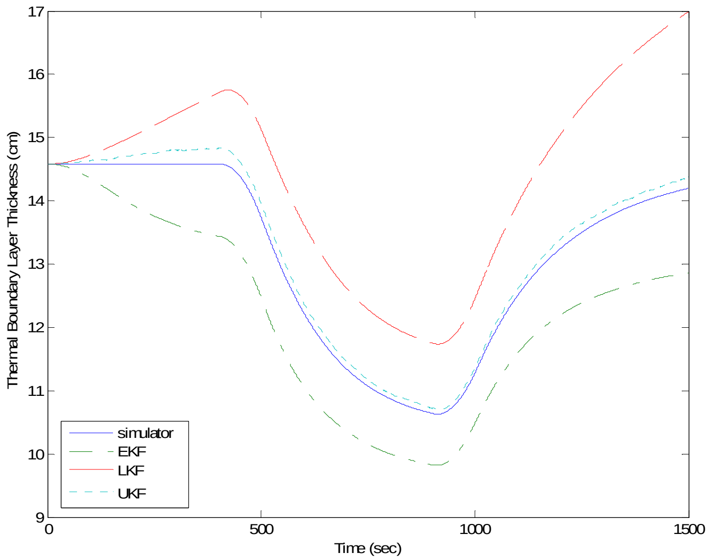


Figure 6.5 Plots of thermal boundary layer thickness by different estimation algorithms

In Figure 6.6, slag temperature estimates show the same trend as in thermal boundary layer thickness. Notice that the slag temperature estimate by the LKF has a negative bias compared to the positive bias in the thermal boundary layer thickness. For immersion depth in Figure 6.7, the results from the UKF look noisy. The EKF shows a small bias but is close to the true value while the estimate from the LKF is diverging.

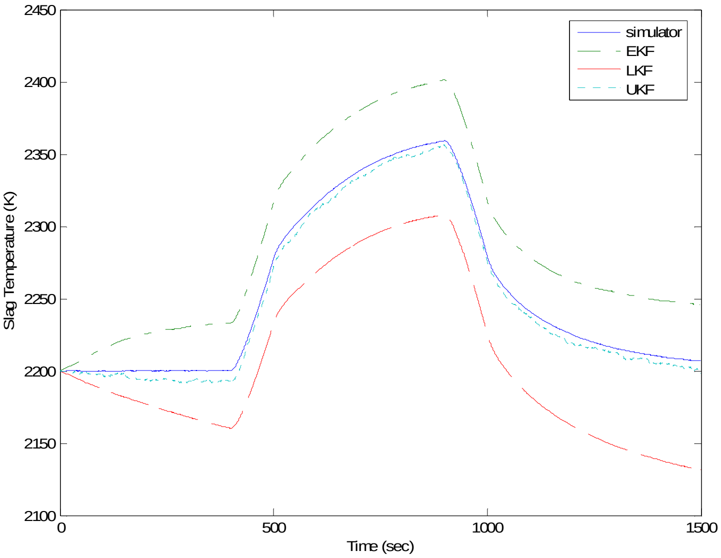


Figure 6.6 Plots of slag temperature by different estimation algorithms

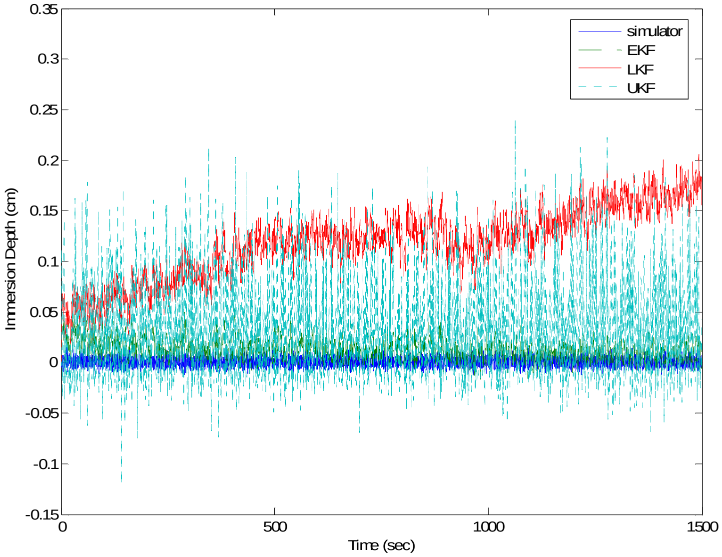


Figure 6.7 Plots of immersion depth by different estimation algorithms

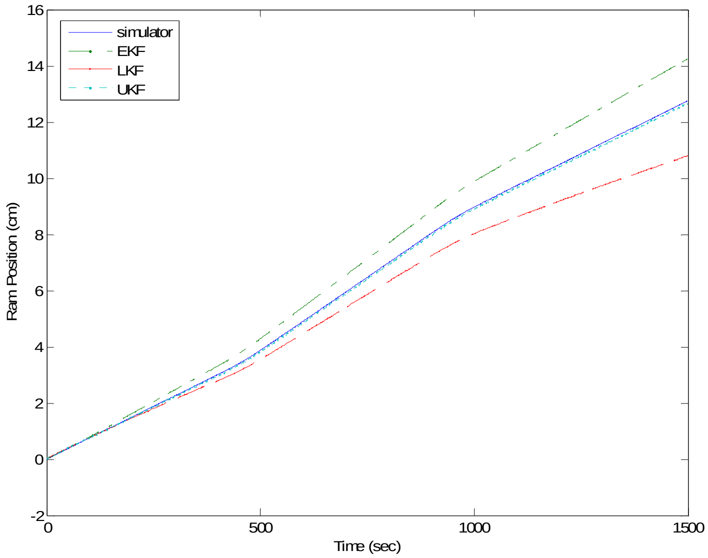


Figure 6.8 Plots of ram position by different estimation algorithms

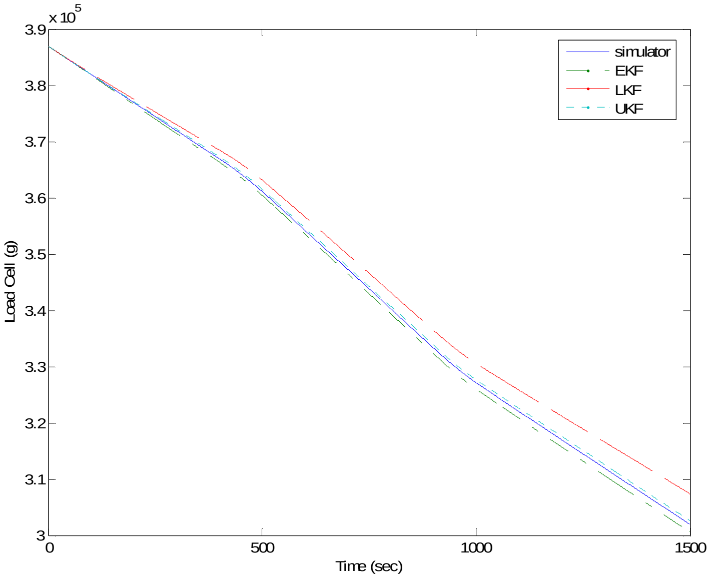


Figure 6.9 Plots of load cell by different estimation algorithms

Figure 6.8 shows the ram position estimate and Figure 6.9 shows the load cell estimate. The estimates by the UKF match well with the true values from the simulator. To compare the estimation performance quantitatively, the average root mean square estimation error for the LKF, EKF, and UKF is calculated and summarized in Table 6.9. The percentage inside the brackets show the relative error reduction percentage for the EKF and UKF compared to the LKF estimation. The UKF estimation is able to reduce estimation error more than 80 % except for the immersion depth compared to the LKF estimation. For immersion depth, the estimate from the EKF shows the best results. We repeated the simulation 100 times for statistical average. The estimates error reduction in thermal boundary layer thickness and slag temperature from the UKF is a big improvement because they normally show a highly nonlinear profile in the simulation.

Table 6.9 Average root-mean-square (RMS) estimation error for LKF, EKF, and UKF (Test 1a)

Algorithm	Average RMS error				
	Thermal boundary layer thickness (cm)	Slag Temperature (K)	Immersion Depth (cm)	Ram Position (cm)	Load Cell (g)
LKF	1.4494	50.2721	0.1222	0.9185	3101.3
EKF	0.9823 (32%)	36.0430 (28%)	0.0147 (88%)	0.8121 (12%)	924.7854 (70%)
UKF	0.1461 (89%)	5.0478 (90%)	0.0534 (56%)	0.0610 (93%)	473.0723 (84%)

Now consider the case where the estimates from the estimators are used to calculate the control inputs. This case will illustrate not only the difference in estimation performance between the each filter but also how that difference in estimates contributes to the overall controller performance. Figure 6.10 shows the melt rate control performance if we use the LKF as the estimation algorithm. Melt rate by the LKF has a bias and its value is higher than the reference melt rate. We can explain the reason why melt rate by the LKF is higher than the given profile if we observe the thermal boundary layer thickness estimate in Figure 6.11. The estimate is higher than the true value from the simulator dynamics. Remembering the causality reasoning control algorithm derivation from equation (5.15) in Chapter 5, the thermal boundary layer thickness term represents heat conduction away from the electrode face while the melt power term accounts for input power to the electrode face. At steady state melting, these two terms are equal which implies that for low power input the boundary layer will be larger at steady state than for high power input. Physically, this implies that at steady state there must be more heat conduction with smaller boundary layer thickness with high power. Since the boundary layer cannot change instantaneously, this means that when the current is ramped up, the boundary layer thickness will be larger than its steady state. This larger than steady state boundary layer yields less heat conduction from the surface and therefore more of the input power goes into melting the electrode until the boundary layer has a time to become shorter. Overestimated thermal boundary layer thickness actually results in increased melt power based on this analysis.

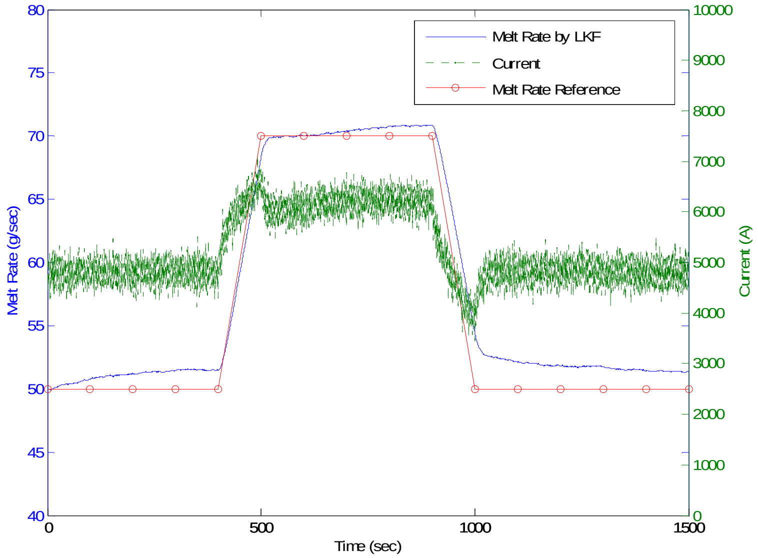


Figure 6.10 Plots of melt rate and input current by LKF (Test 1a)

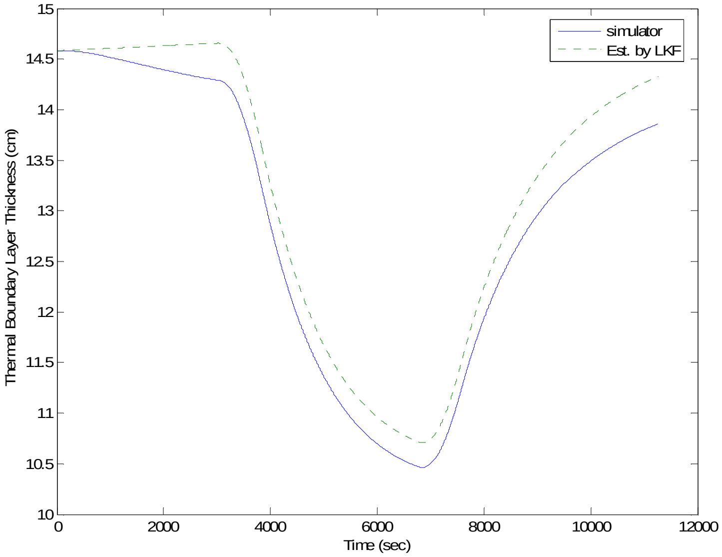


Figure 6.11 Plots of thermal boundary layer thickness by LKF (Test 1a)

. The opposite case can be observed when the EKF is being used as the estimator algorithm. The melt rate control performance and the thermal boundary layer thickness estimation by the EKF plots are shown in Figure 6.12 and Figure 6.13. The melt rate control performance and current input using the UKF as the estimator for control is shown in Figure 6.14. The thermal boundary layer thickness estimation is shown in Figure 6.15. The melt rate estimate and the thermal boundary layer thickness estimate from the UKF match well with the simulator value.

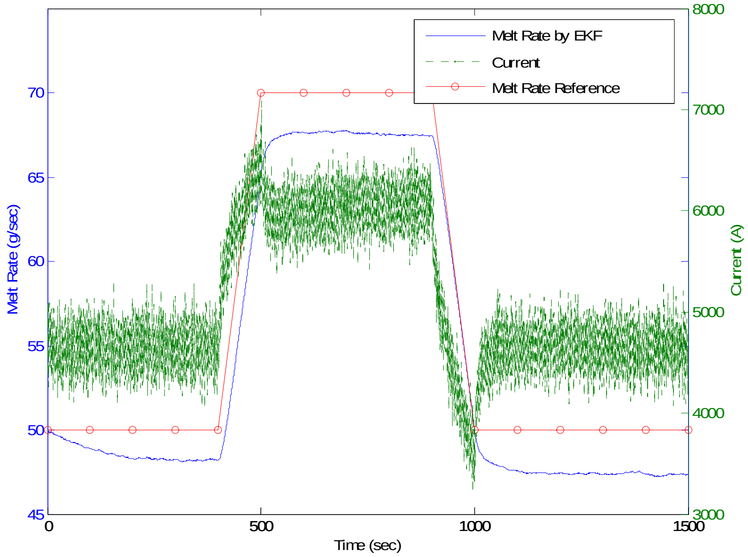


Figure 6.12 Plots of melt rate and input current by EKF (Test 1a)

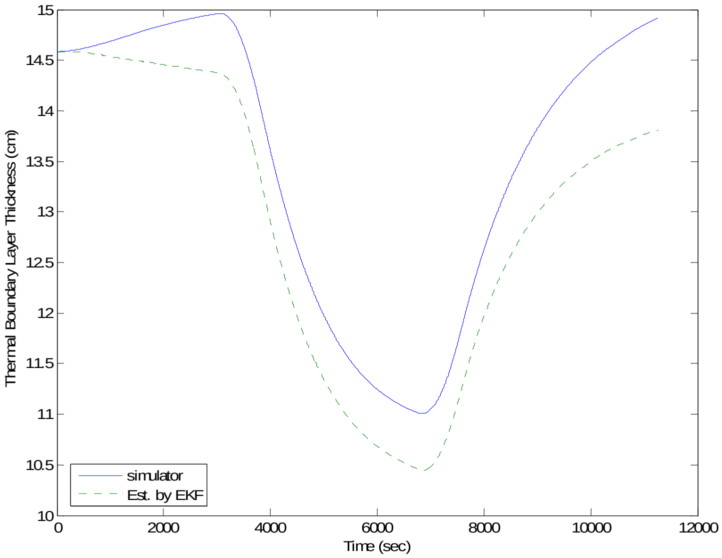


Figure 6.13 Plots of thermal boundary layer thickness by EKF (Test 1a)

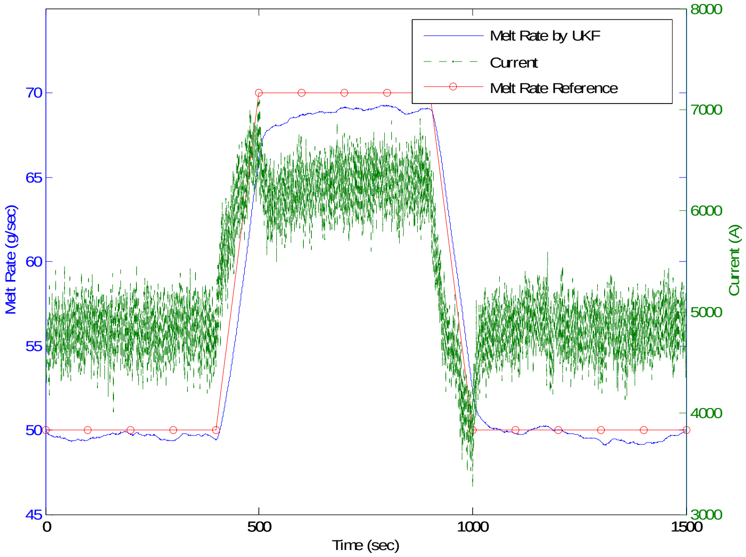


Figure 6.14 Plots of melt rate and input current by UKF (Test 1a)

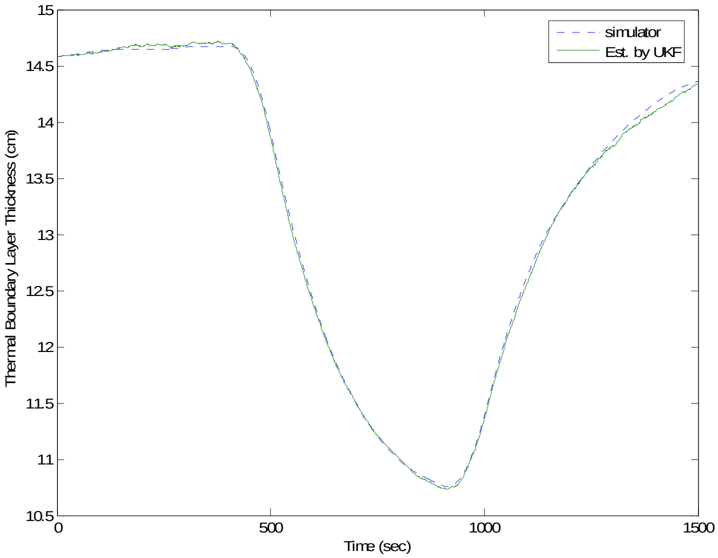


Figure 6.15 Plots of thermal boundary layer thickness by UKF (Test 1a)

To verify the control performance of immersion depth and the immersion depth as a function of voltage signals in measurement dynamics, we made a modified melt rate profile with a varying immersion depth profile. For this case the UKF estimation algorithm was used as an estimator. The modified melt rate reference and immersion depth profile is shown in Figure 6.16. Figure 6.17 shows the immersion depth control performance and ram velocity change as a function of time. Notice that the ram velocity is responding to compensate for errors in immersion depth. Input current does not show noticeable response to the immersion depth variation for this case as shown in Figure 6.18. The suggested model for estimating the immersion depth as a function of voltage signals works well showing the measured impedance responds with immersion depth variation as shown in Figure 6.19.

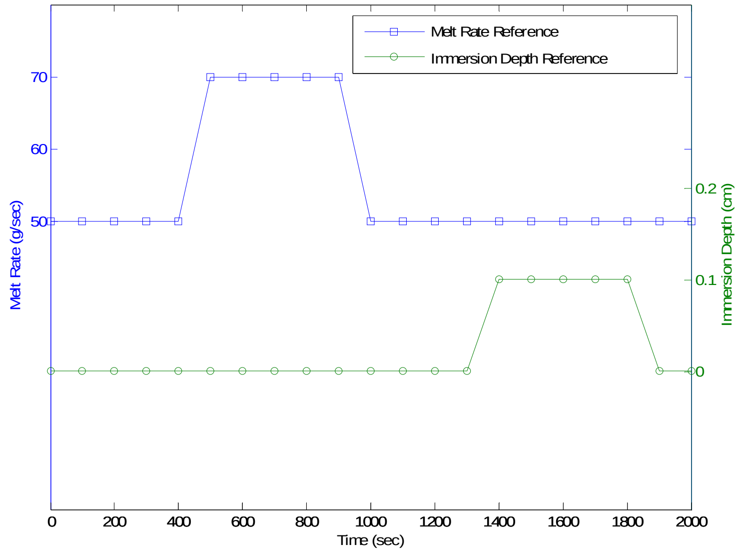


Figure 6.16 Plots of melt rate reference and immersion depth reference used in Test 1a for immersion depth control performance verification

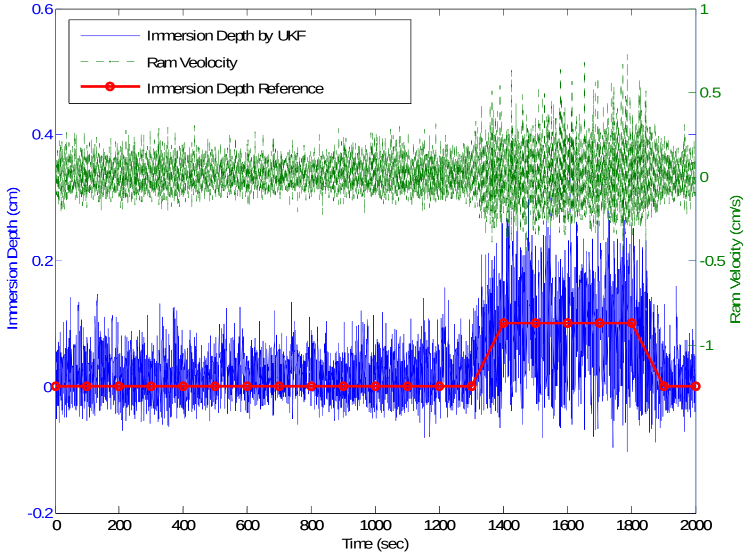


Figure 6.17 Plots of immersion depth and ram velocity by UKF for immersion depth control performance verification (Test 1a)

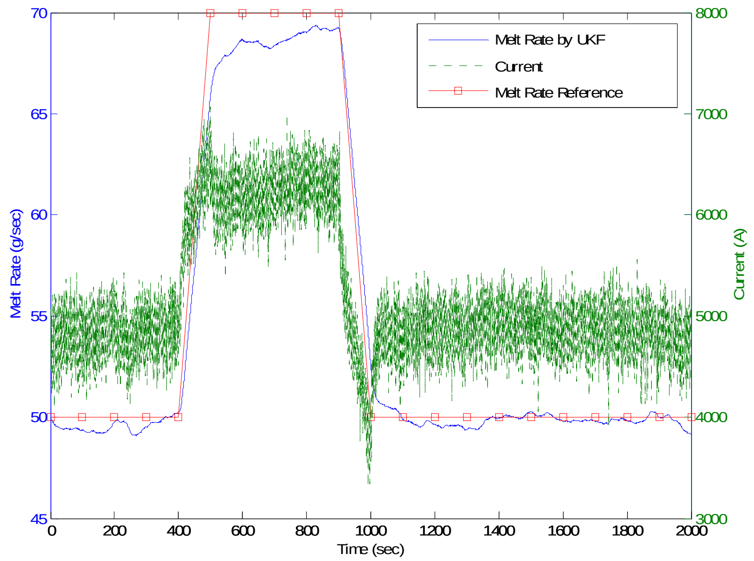


Figure 6.18 Plots of melt rate and input current by UKF for immersion depth control performance verification (Test 1a)

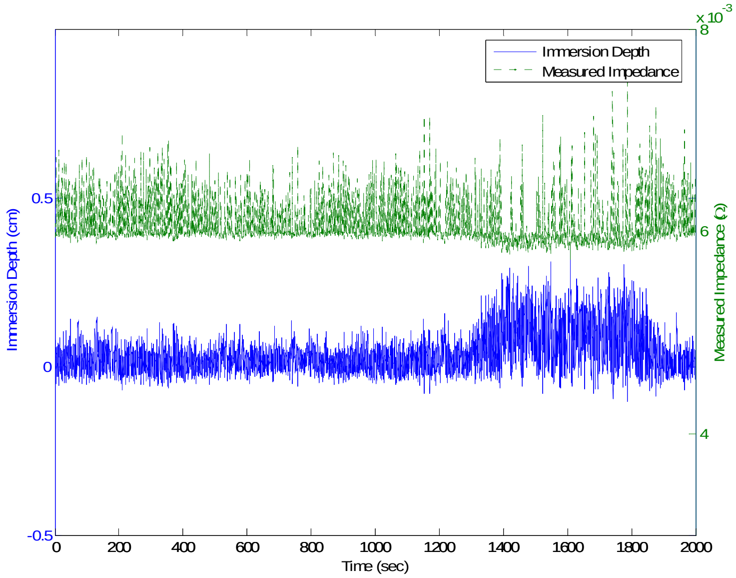


Figure 6.19 Plots of immersion depth and measured impedance by UKF for immersion depth control performance verification (Test 1a)

The performance of the particle filter as an estimator for the controller is compared with the UKF counterpart. We used two different noise profiles: Gaussian and Non-Gaussian. For the Gaussian case, we generated artificial noise as in the previous test. The Non-Gaussian noises were generated based on the Gamma function shown in Figure 6.20.

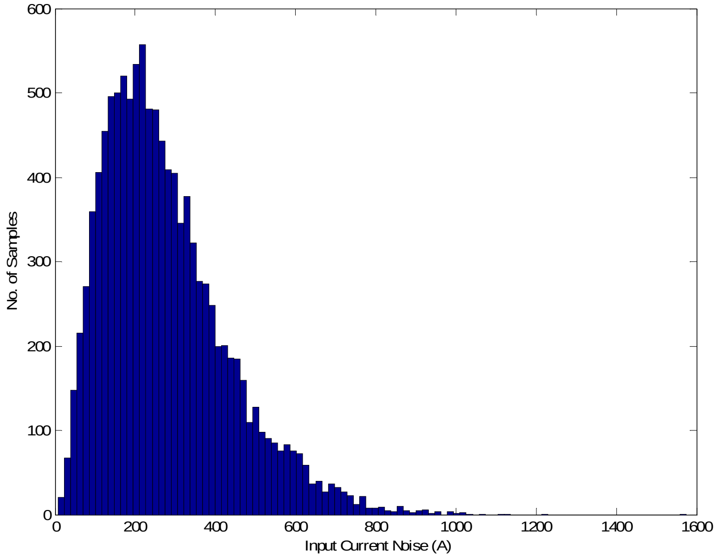


Figure 6.20 Histogram of input current noise generated based on Gamma distribution for non-Gaussian noise case

Due to the characteristics of the Monte Carlo technique, more particles produce better results. For Test 1a, the sampling time is 0.133 seconds. If the particle filter can calculate the estimate for each time step in 0.133 seconds; we might consider it as appropriate for real-time implementation. The Matlab code is programmed based on the "easy-to-understand" philosophy. We also save many of the filter variables for plotting purposes and later analysis, so this constraint is very weak. The LKF, EKF, and UKF based control designs satisfy this requirement. For the particle filter, the number of particles based on the above constraint is very small. So here, we just use enough particles to get an adequate result. 200 particles are used to calculate the sequential importance resampling based particle filter. The results are summarized in Table 6.10.

Table 6.10 Average root-mean-square (RMS) estimation error for particle filtering and UKF (Test 1a) in the control system

Algorithm	Average RMS error				
	Thermal boundary layer thickness (cm)	Slag Temperature (K)	Immersion Depth (cm)	Ram Position (cm)	Load Cell (g)
UKF	0.1461	5.0478	0.0534	0.0610	473.0723
PF	0.1499	5.8203	0.0871	0.0668	597.18
UKF (Gamma)	0.5951	23.66	0.2484	0.3396	1747.5
PF (Gamma)	0.445	17.189	0.4398	0.1514	1784.7

For the Gaussian noise case, the performances of the particle filter and the UKF are almost identical except the immersion depth estimate. It means the accuracy of the particle filter easily surpasses the LKF and EKF. For the Non-Gaussian noise case, the particle filter based control design produced less error than the UKF counterpart except in the immersion depth case. The average calculation time for particle filter to estimate each time step is about 1.2 seconds (from the 1.6 Ghz Pentium M processor with 512 Mbytes memory based laptop).

This computational burden prevents us from further investigation of using the particle filter as an estimator for the ESR process control design. Further research is needed to reduce the particle size without compromising the estimator performance.

6.2 TEST 1B: LINEAR RAMP MELT PROFILE WITH ALL OTHER NOISES

All other conditions for the Test 1b are identical to the conditions for Test 1a except the process noise covariance. The process noise covariance matrix **Q** becomes

$$\mathbf{Q} = \begin{bmatrix} {\sigma_{I_c}}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & {\sigma_{Vram}}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t_s{\sigma_{\mu_r}}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & t_s{\sigma_a}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_s{\sigma_{V_b}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t_s{\sigma_{I_b}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t_s{\sigma_{V_{ram_b}}}^2 \end{bmatrix} \tag{6.3}$$

Table 6.11 and 6.12 contain a list of the noise strength used in this simulation. The details of the implementation procedure for the simulation algorithm are explained as a Matlab pseudo-code in Appendix C.

Table 6.11 Input Noise strength

Parameter	Description	Values
σ_I	Current input standard deviation	184.2 (A)
σ_{Vram}	Ram speed standard deviation	0.01 (cm/sec)

Table 6.12 Process Modeling noise Strength

Parameter	Description	Values
σ_{μ_r}	Melt efficiency standard deviation	0.0005
σ_a	Fill ratio parameter standard deviation $(0.01a_0)$	0.0036
σ_{Ib}	Bias current standard deviation $0.0005I_0$	2.566 (Ohm)
σ_{Vramb}	Bias ram speed standard deviation $0.1Vram_0$	$7.50*10^{-4}$ (cm/sec)
σ_{Voltb}	Bias Voltage standard deviation $0.0005V_0$	0.0138 (V)

For estimator's performance comparison, the thermal boundary layer thickness estimation result is shown in Figure 6.21. It shows the same trend as in Test 1a, but the LKF and EKF estimations of the thermal boundary layer thickness show larger errors than in the Test 1a case.

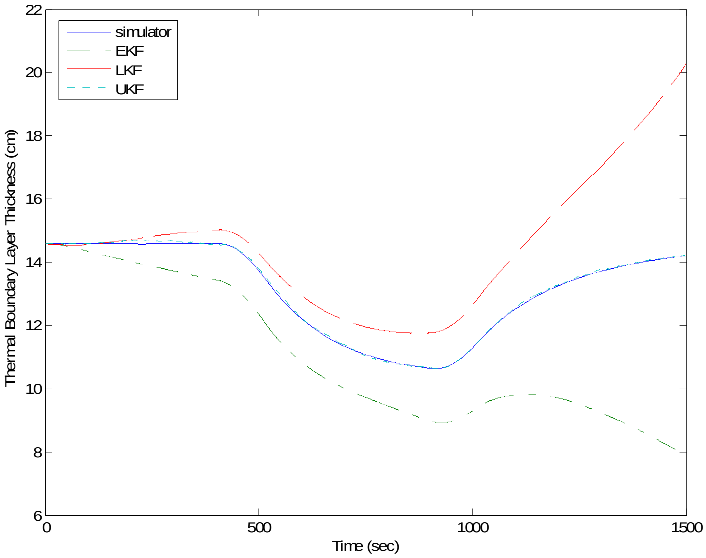


Figure 6.21 Plots of thermal boundary layer thickness by different estimation algorithms (Test 1b)

The estimation by the UKF matches well with the simulator even with the increased uncertainties. In Table 6.13, average RMS error for the estimation is summarized in the same way as Table 6.9. For Test 1b, relative melt efficiency, area fill ratio, voltage bias, current bias, ram velocity bias are treated as disturbance variables. Except for the immersion depth and ram position, the performance of the UKF is 80% more accurate than the LKF estimates. Additionally, the UKF is able to reduce the thermal boundary layer thickness errors by 99% and is the biggest improvement over the LKF and EKF. From the Test 1a we have learned that the accuracy of thermal boundary layer thickness estimation is directly related with the performance of the melt rate control.

Table 6.13 Average root-mean-square (RMS) estimation error for LKF, EKF, and UKF (Test 1b)

Algorithm	Average RMS error				
	Thermal boundary layer thickness (cm)	Slag Temperature (K)	Immersion Depth (cm)	Ram Position (cm)	Load Cell (g)
LKF	3.886	4.9624	0.0270	0.0708	1526.5
EKF	2.0475 (47%)	27.449 (-82%)	0.0150 (44%)	0.0523 (26%)	1054.7 (31%)
UKF	0.0404 (99%)	0.8678 (83%)	0.0514 (-47%)	0.1072 (-34%)	56.234 (96%)
Algorithm	Average RMS error				
	Relative Melt Efficiency	Area Fill Ratio	Voltage Bias (V)	Current Bias (A)	Ram Velocity Bias (cm/s)
LKF	0.2175	0.1332	2.3428	257.31	0.0222
EKF	0.1437 (34%)	0.0717 (46%)	1.1324 (52%)	265.07 (-3%)	0.0214 (4%)
UKF	0.0066 (97%)	0.0959 (28%)	0.3245 (86%)	26.049 (90%)	0.0042 (81%)

Now investigate the case where the estimates from the estimators are used to calculate the control inputs. Figure 6.22 shows the melt rate control performance if we use the LKF as the estimation algorithm. More than 60% of simulations for this case ended in failure. We believe that the increasing error in the thermal boundary layer thickness guides the control inputs to the wrong values and causes the whole system simulation to diverge.

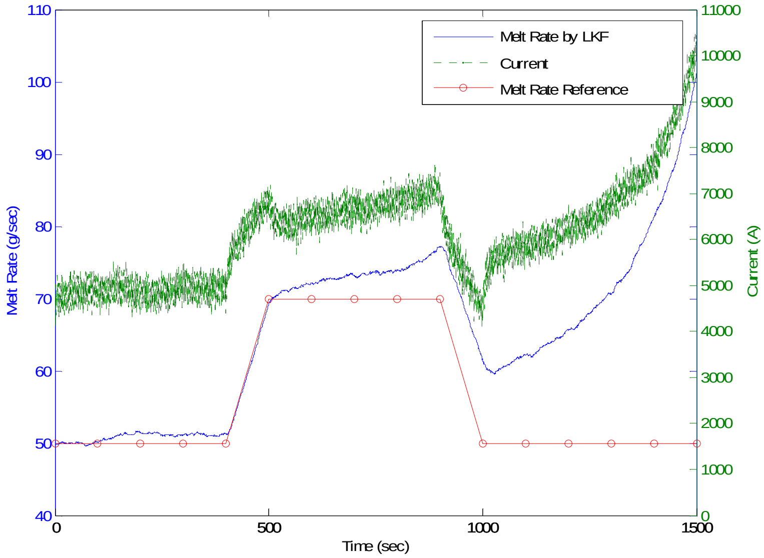


Figure 6.22 Plots of melt rate and input current by LKF (Test 1b)

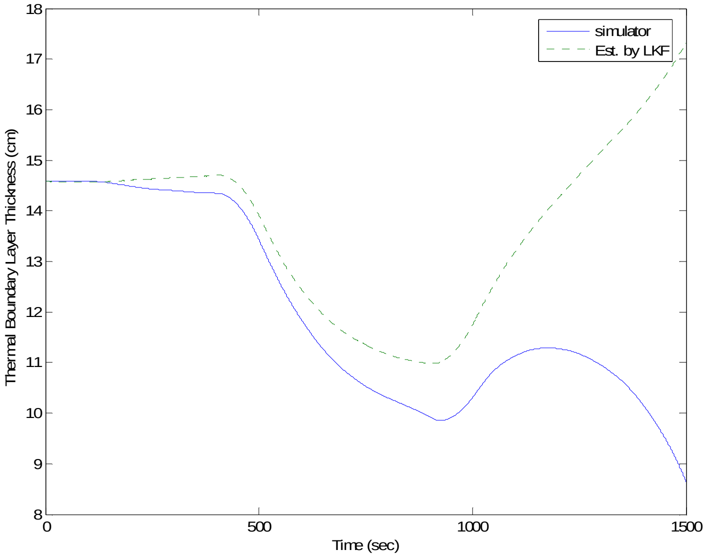


Figure 6.23 Plots of thermal boundary layer thickness by LKF (Test 1b)

The thermal boundary layer thickness estimation result is shown in Figure 6.23. The same reasoning can be applied as in Test 1a case. Figure 6.24 shows the melt rate control performance and current input using the EKF as the estimator for control. This is the same trend as was shown in Test 1a, but for this simulation it diverges at the end. The related thermal boundary layer thickness estimation is shown in Figure 6.25. The thermal boundary layer thickness diverges after the current ramp down. Figure 6.26 shows the melt rate control performance by the UKF. The melt rate is very close to the reference melt rate profile. The thermal boundary layer thickness estimation is shown in Figure 6.27. The estimation of the thermal boundary layer thickness is very accurate compared to the estimates from the LKF and EKF. The immersion depth control resulting from the EKF is shown in Figure 6.28. The results for the immersion depth control using the UKF is shown in Figure 6.29. In either case, the immersion depth control performance is in an acceptable range. From the above results and the average RMS estimation error comparison in Table 6.13, our choice for the estimator is the UKF estimation algorithm.

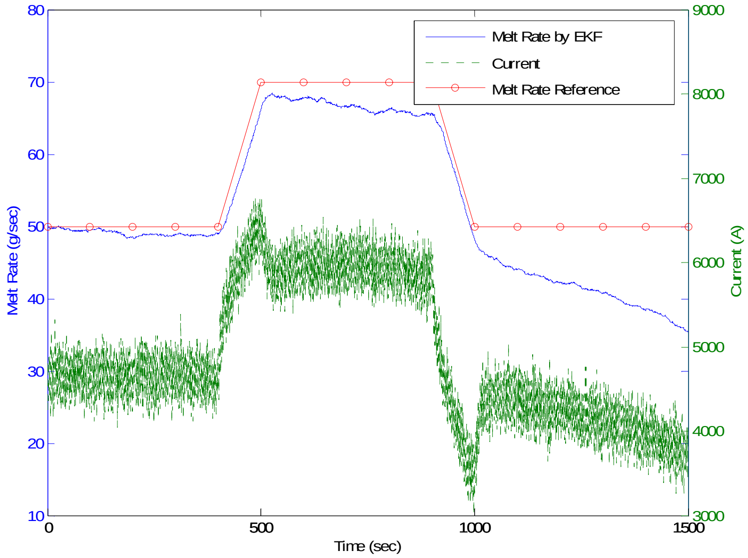


Figure 6.24 Plots of melt rate and input current by EKF (Test 1b)

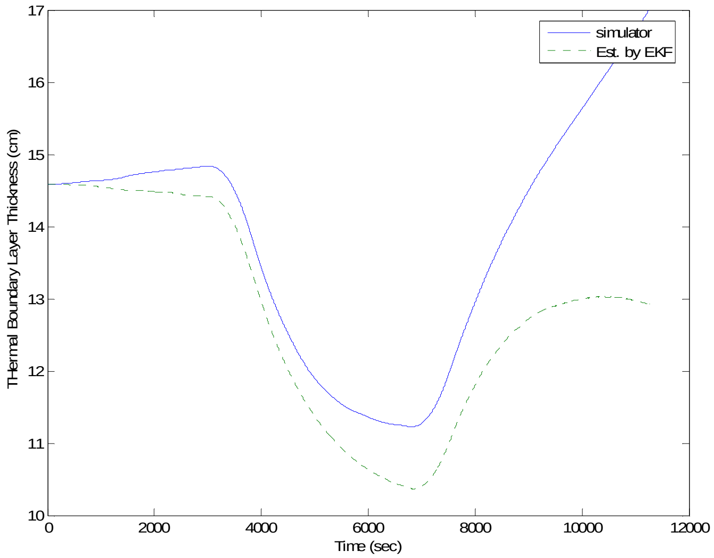


Figure 6.25 Plots of thermal boundary layer thickness by EKF (Test 1b)

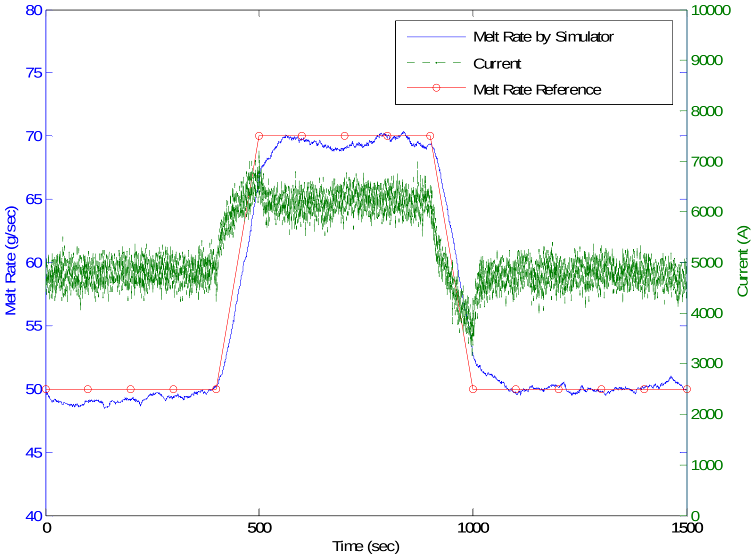


Figure 6.26 Plots of melt rate and input current by UKF (Test 1b)

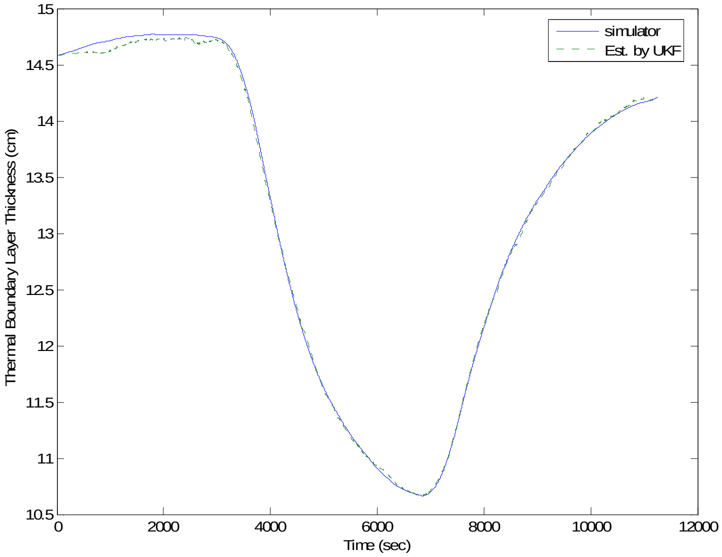


Figure 6.27 Plots of thermal boundary layer thickness by UKF (Test 1b)

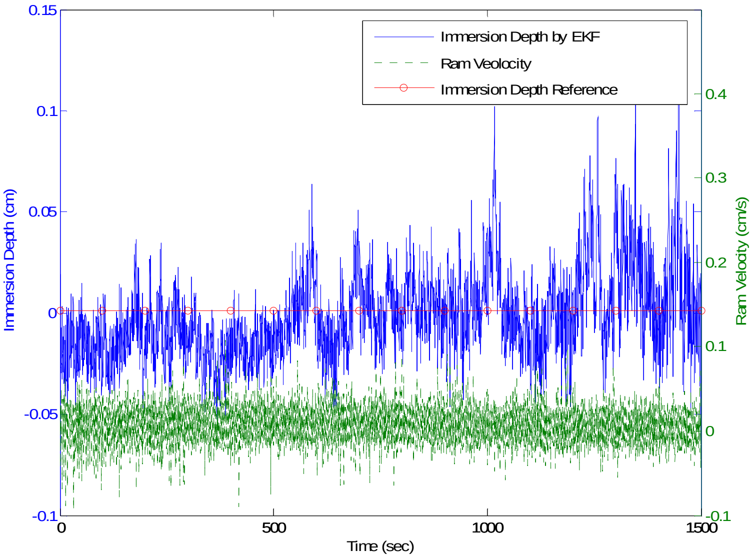


Figure 6.28 Plots of immersion depth and ram velocity by EKF for immersion depth control performance verification (Test 1b)

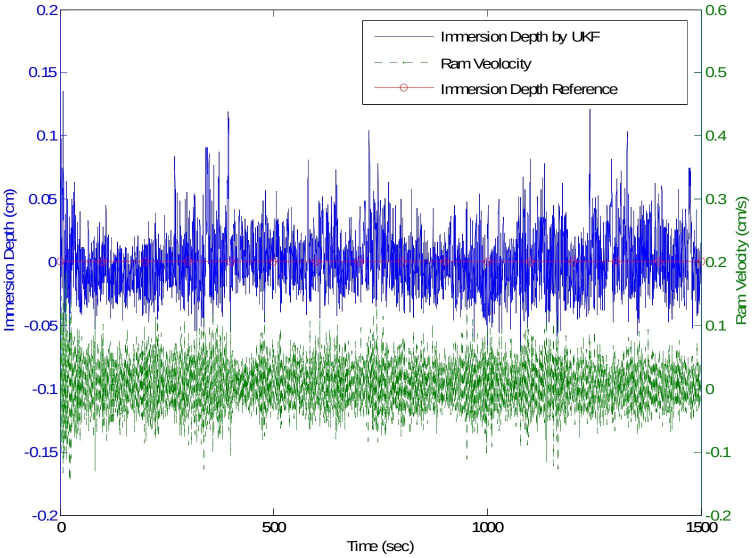


Figure 6.29 Plots of immersion depth and ram velocity by UKF for immersion depth control performance verification (Test 1b)

6.3 TEST2: PROFILE BEING USED IN THE INDUSTRY

In a typical melt, the first stage is a controlled build-up of power at the prescribed rate, to a peak just over the maximum required. This is tolerable at this early stage of the melt, where base plate cooling still has an effect. Current is then reduced to a level that will give the desired melt rate and conditions are monitored relative to the reference program. It is usually preferable to isolate any automatic controller that could cause violent electrode withdrawal at this early stage possibly resulting in complete loss of electrode contact and failure to re-establish contact if slag material fills the gap [20]. Our simulation is based on the assumption that we turn on the controller after the current is reduced to a level that will give the desired melt rate. One of the examples for a traditional melt profile from the industry (courtesy of Consarc Corporaton) is shown in Figure 6.30 [20]. The melt rate starts at 20 (g/sec) and initial slag temperature is set at 2000 (K). The melt rate control performance is good and does not show any overshoot or undershoot in Figure 6.31. The average RMS error for melt rate is about 0.5 (g/sec). The current input is increasing continuously after the melt rate reached the target melt rate. This is due to relatively slow responding thermal boundary layer thickness growth. Figure 6.32 shows the thermal boundary layer thickness and input current. The thermal boundary layer thickness reaches its steady state for a given melt rate about at 1600 seconds. Until *t*=1600 seconds the melting dynamics do not reach steady state and nonlinear profile control inputs will be needed. The thermal boundary layer thickness is larger than expected for steady state. As a result, a continuous increase of input current is needed.

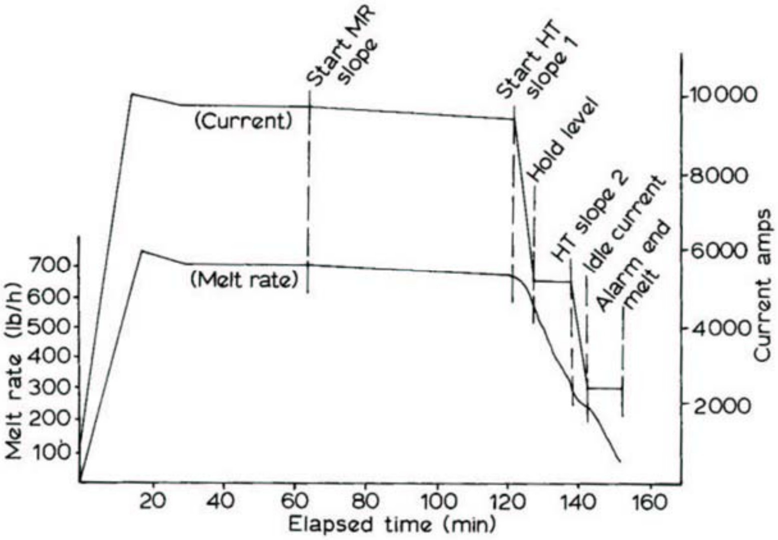


Figure 6.30 Typical melt profile used in the industry [20]

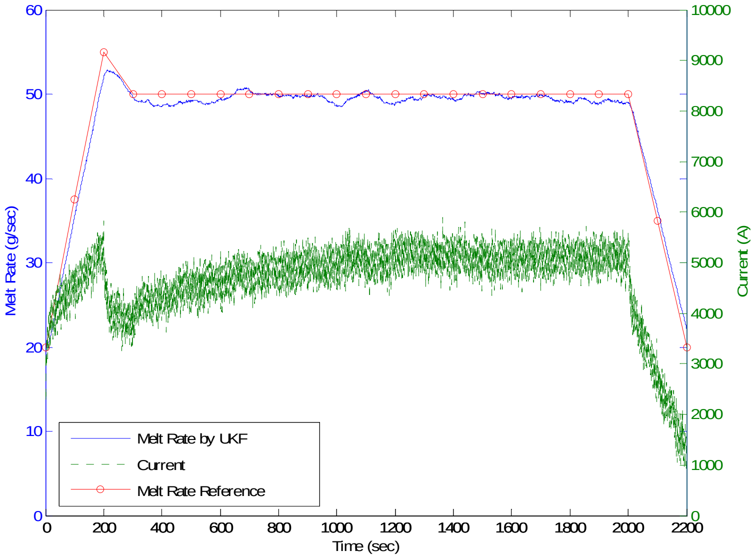


Figure 6.31 Plots of melt rate and input current by LKF (Hoyle)

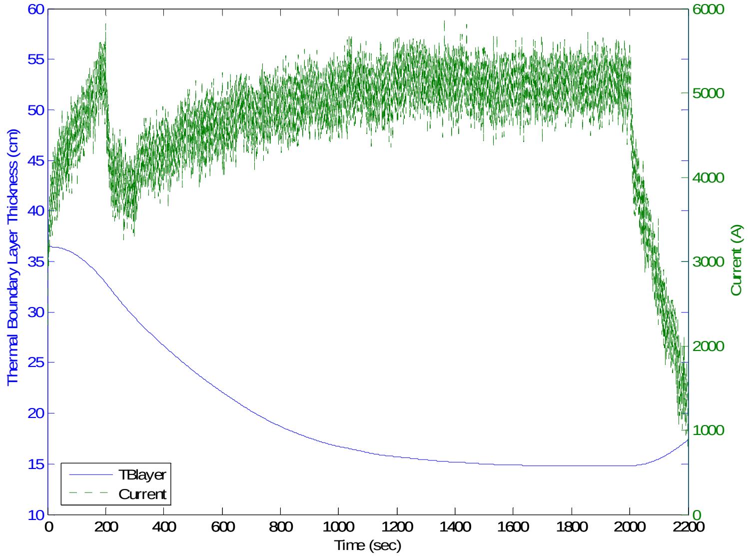


Figure 6.32 Plots of thermal boundary layer thickness and input current of Hoyle's melt rate profile

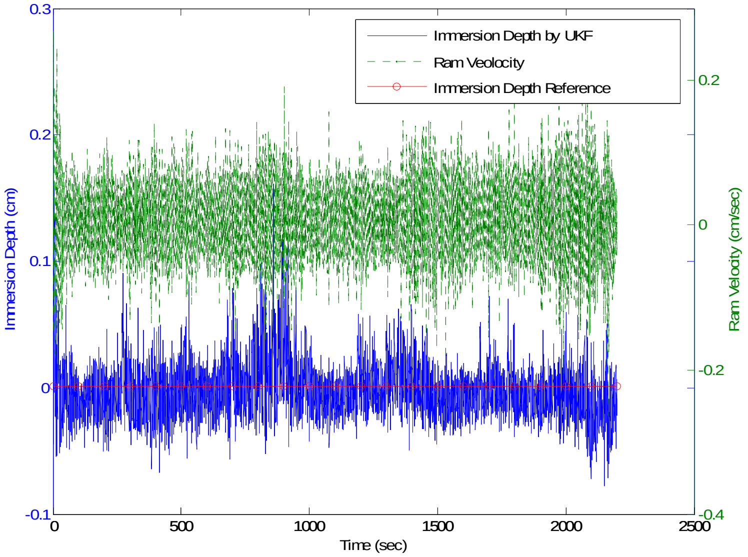


Figure 6.33 Plots of immersion depth and ram velocity (Hoyle)

This explains the sudden drop in input current at *t*=200 seconds. The melt rate reached the desired level but at the tip of the electrode heat is building up because of the thermal boundary layer propagation. As a result, the input current has to be decreased.

With the given initial conditions, the system reaches steady state at about 1600 seconds. The suggested control design works well in this transient period and achieves the target melt profile without significant errors. Figure 6.33 shows the immersion depth control performance and ram velocity control input. The average RMS error of immersion depth is 0.0255 (cm) and is in the acceptable range for industrial applications.

6.4 DISCUSSION

The results of the melt rate simulation of Test 1a show that relatively steep melt rate ramps can be accurately implemented with no significant over or undershoot of the end melt rate targets if we have perfect estimates. We applied three estimation algorithms into the ESR control design. Test 1 used a linear ramp melt profile to investigate the suggested control design in transient conditions. Two different noise dynamics conditions are used. First, only the control input noises are considered as process noises. The estimation performance for state variables by the LKF and EKF showed biased estimates and errors. The estimation performance of the UKF is good and compared to the LKF the relative errors are reduced more than 90 % for the thermal boundary layer thickness estimate. The control performance as a whole system showed the same trends as in the estimation performance, and we have learned that the error in the thermal boundary layer thickness directly affects the melt rate performance. If we increase the system dynamics' uncertainties by considering more process noise terms, the control design with the LKF as an estimator fails to produce acceptable results about 60 percent of the time. For the successful cases, the melt rate performance is very poor and the results diverge. For the EKF, the simulation runs stable, but the control performance is very poor for melt rate control. The performance of the control design with the UKF as an estimator is very impressive compared to the LKF and EKF. So we chose the UKF as the estimator for control design and applied the melt profile which being used in the industry. The results from the suggested control design show that the controller automatically and correctly compensated for the perturbations when the electrode thermal distribution was driven away from the steady state as in start-up and hot-top processes.

CHAPTER 7

CONCLUSIONS

The description of the electroslag remelting process and the motivation for modeling, estimation and control of the process were introduced in Chapter 1. A model based, observer based control design was suggested. As possible candidates for the observer design, the Kalman filter, extended Kalman filter, unscented Kalman filter and particle filter were introduced. The goal of this dissertation was defined as *Given desired values for the melt rate and immersion depth, determine the time varying current and ram speed inputs required to establish and maintain these desired outputs under noisy environments.*

Chapter 2 provided a reduced order model of the electroslag remelting process. The electrode melting dynamics is a Stefan problem with a moving phase boundary. The model was derived based on the heat ablation model with boundary layer approximation and the integral method. The energy balance equation was used to derive the heat transfer model of the slag. The suggested model was experimentally verified.

Chapter 3 provided an in-depth review of recursive Bayesian estimation theory. As an optimal recursive Bayesian estimation algorithm, the Kalman filter was introduced. To deal with nonlinear and non-Gaussian problems, suboptimal recursive Bayesian estimation algorithms were introduced. The extended Kalman filter and its limitations from using the 1st order linearization were discussed. The unscented Kalman filter was introduced as an alternative. By using the unscented transformation, we showed that the UKF can exactly estimate the mean and covariance up to the 2nd order of the Taylor series.

In Chapter 4, a Monte Carlo based approach for filtering problems was introduced. Unlike Gaussian approximation solutions such as the EKF and UKF, the Monte Carlo based recursive estimation algorithm using the sequential importance resampling can deal with any nonlinearity and non-Gaussianity.

In Chapter 5, we focused on design of the controller and estimator based on the previous chapters for the ESR process. By using the causality reasoning, a nonlinear controller has been derived. To estimate the related state variables the LKF, EKF, UKF and particle filter based estimators were designed.

Chapter 6 covered the performance evaluation results from the suggested control design. The linear ramp based melt rate profile and two different noise dynamics were used. The estimation performance of the UKF is better than the LKF or an un-tuned EKF. The control performance using these estimation algorithms was also investigated. The performance of the particle filter was compared with the results from the UKF. Our choice was the UKF. By applying the UKF based control design to a typical melt rate profile used in the industry, the applicability of the suggested design has been verified.

The following conclusions may be drawn from this dissertation:

1. A reduced order model of the electroslag remelting process has been successfully derived and proved to be useful in control

applications. The integral methods were used to derive a set of coupled nonlinear ordinary differential equations in time, which capture the steady state and transient characteristics of melting in the ESR process.

2. An accurate, model-based melt rate and immersion depth control for the electroslag remelting process was developed. The suggested control algorithm is based on the premise that physical understanding of the process should be combined with measurement to provide an optimal estimate of the quantities to be controlled. Therefore, the quality of the control is directly related with the quality of the observer performance.
3. The unscented Kalman filter was chosen for the observer in the control design, and the control system is capable of implementing steep, linear melt rate ramping without significant overshoot or undershoot of the final target melt rate. The suggested controller correctly senses and correctly responds to the disturbances caused by uncertainties in the melting dynamics, geometry change, bias in the current and voltage, and bias in the ram speed.
4. The controller utilizing the linearized Kalman filter or the un-tuned version of extended Kalman filter show a bias in the melt rate control performance or even diverge if the disturbances in the melting dynamics, geometry change, ram velocity bias and the biases in the voltage signals are considered.

5. The controller utilizing the particle filter shows good performance compared to the UKF especially under the non-Gaussian noise environment. Its heavy computational burden, use in the ESR control applications is not feasible at the present time.

APPENDIX A

DERIVATION OF MELTING DYNAMICS OF THE ELECTRODE

Based on the Zien [54] and Vujavovic's solution [53, 55] for the heat ablation, the melting dynamics of the ESR problem can be solved as follows.

In order to use the approximate integral method, it is convenient to transform heat conduction equation (2.5) shown in Chapter 2 into two equations as

$$q = -\alpha(h)\frac{\partial h}{\partial x} \tag{A.1}$$

And the related energy conservation equation is

$$\frac{\partial h}{\partial t} = -\frac{\partial q}{\partial x} \tag{A.2}$$

where h is the volume specific enthalpy relative to the room temperature defined as

$$h = \int_{T_r}^{T} \rho C dT \tag{A.3}$$

where q is heat flux in the electrode, and α is temperature specific thermal diffusion coefficient which is

$$\alpha = \frac{k}{\rho C} = \alpha(h) \tag{A.4}$$

Here we only assume temperature dependency of α rather than separate (ρ , C , k) in (2.5). The modified boundary conditions with boundary layer approximations are

$$h(S_e + \Delta) = 0 \Leftarrow T(S_e + \Delta) = T_r \tag{A.5}$$

$$q(S_e + \Delta) = 0 \Leftarrow \frac{\partial T}{\partial x}(S_e + \Delta) = 0 \tag{A.6}$$

where Δ is an unknown but finite boundary layer thickness (cm) in which the temperature is assumed to be at room temperature and the heat flux is zero. If we integrate equation (A.2) from the melt front S_e to the boundary layer $S_e + \Delta$ (Refer Figure 2.2 for the coordinate setup)

$$\begin{aligned} \int_{S_e}^{S_e+\Delta} \frac{\partial h}{\partial t} dx &= -\int_{S_e}^{S_e+\Delta} \frac{\partial q}{\partial x} dx \\ &= -q(S_e + \Delta) + q(S_e) \end{aligned} \tag{A.7}$$

If we apply the boundary condition at the melt front to the equation (A.7), then

$$\frac{P_m(I,d)}{A_e} - \rho_{\text{sup}} L^* \dot{S}_e = \int_{S_e}^{S_e+\Delta} \frac{\partial h}{\partial t} dx \tag{A.8}$$

Assume an enthalpy profile as

$$h = h_m \left(1 - \eta\right)^3 \qquad \text{(A.9)}$$

where $\eta = \dfrac{x - S_e}{\Delta}$ which satisfies the following conditions

At $x = S_e + \Delta$ and $x = S_e$,

$$h(S_e + \Delta) = 0 \qquad \text{(A.10)}$$

$$h(S_e) = h_m \qquad \text{(A.11)}$$

$$T(S_e) = T_m \qquad \text{(A.12)}$$

To integrate the equation (A.8) we expand the derivative term $\dfrac{\partial h}{\partial t}$ as

$$\begin{aligned}
\left.\frac{\partial h}{\partial t}\right|_x &= \left.\frac{\partial h}{\partial t}\right|_\eta + \frac{\partial h}{\partial \eta}\left(\frac{\partial \eta}{\partial S_e}\dot{S}_e + \frac{\partial \eta}{\partial \Delta}\dot{\Delta}\right) \\
&= \left.\frac{\partial h}{\partial t}\right|_\eta + \frac{\partial h}{\partial \eta}\left(\frac{-1}{\Delta}\dot{S}_e - \frac{x - S_e}{\Delta^2}\dot{\Delta}\right) \\
&= \left.\frac{\partial h}{\partial t}\right|_\eta - \frac{\partial h}{\partial \eta}\left(\frac{\dot{S}_e}{\Delta} + \frac{\eta}{\Delta}\dot{\Delta}\right) \\
&= \left.\frac{\partial h}{\partial t}\right|_\eta - \frac{1}{\Delta}\frac{\partial h}{\partial \eta}\left(\dot{S}_e + \eta\dot{\Delta}\right)
\end{aligned} \qquad \text{(A.13)}$$

Then by applying equation (A.9), the equation (A.13) becomes

$$\left.\frac{\partial h}{\partial t}\right|_x = \frac{h_m}{\Delta} 3(1-\eta)^2 \left(\dot{S}_e + \eta\dot{\Delta}\right) \tag{A.14}$$

Put equation (A.14) into the melt front boundary condition (A.8) then we have

$$p_m - \rho_{\text{sup}} L^* \dot{S}_e = \int_{S_e}^{S_e+\Delta} \frac{\partial h}{\partial t} dx = \int_0^1 \frac{\partial h}{\partial t} \Delta d\eta \tag{A.15}$$

$$p_m - \rho L^* \dot{S}_e = \int_0^1 h_m 3(1-\eta)^2 \left(\dot{S}_e + \eta\dot{\Delta}\right) d\eta \tag{A.16}$$

where

$$p_m = \frac{\mathrm{P_m}}{\mathrm{A_e}} : \text{melting power flux (W/cm}^2\text{)}$$

$$\int_0^1 (1-\eta)^2 d\eta = \frac{1}{3} \tag{A.17}$$

$$\int_0^1 (1-\eta)^2 \eta d\eta = \frac{1}{12}$$

Applying the constants calculated from the above equations into (A.16) gives

$$p_m - \rho L^* \dot{S}_e = h_m \dot{S}_e + \frac{1}{4} h_m \dot{\Delta} \tag{A.18}$$

$$p_m = \left(h_m + \rho L^*\right) \dot{S}_e + \frac{1}{4} h_m \dot{\Delta} \tag{A.19}$$

$$p_m = h_{\text{sup}} \dot{S}_e + \frac{1}{4} h_m \dot{\Delta} \tag{A.20}$$

where, $h_{\sup}$ is enthalpy changes from T_r to $T_{\sup}$. Equation (A.20) represents one equation that needs to be solved for the two unknown function of time S_e and Δ.

To solve for the burn off rate and the thermal boundary layer thickness, it is necessary to develop one more equation relating the two variables. The two equations can then be solved simultaneously. Zien [54] and Vujanovic [53, 55] suggested a moment integral technique rather than the simple match at $x = S_e$ boundary for $\alpha \frac{\partial h}{\partial x}$. This can be performed by rewriting the equation (A.1) and (A.2) as

$$\frac{\partial H}{\partial x} = -h \tag{A.21}$$

$$\frac{\partial H}{\partial t} = -\alpha \frac{\partial h}{\partial x} \tag{A.22}$$

where H is called the heat displacement function and is a negative x integral of h. We take the product of (A.21) and (A.22) then integrate

$$\int_{S_e}^{S_e+\Delta} \alpha h \frac{\partial h}{\partial x} dx = \int_{S_e}^{S_e+\Delta} \frac{\partial H}{\partial x} \frac{\partial H}{\partial t} dx \tag{A.23}$$

The diffusivity will be assumed to vary linearly with enthalpy change.

$$\alpha = \alpha_r (1 + \beta h) \tag{A.24}$$

where

$$\beta = \frac{\alpha_m - \alpha_r}{\alpha_r} \tag{A.25}$$

Using equation (A.24) and (A.25), the equation (A.23) can be expressed as

$$\int_{S_e}^{S_e+\Delta} \alpha h \frac{\partial h}{\partial x} dx = \int_{S_e}^{S_e+\Delta} \alpha h \frac{\partial h}{\partial \eta} d\eta \tag{A.26}$$

$$= \int_{S_e}^{S_e+\Delta} \alpha_r (1+\beta h) h \frac{\partial h}{\partial \eta} d\eta \tag{A.27}$$

$$= \alpha_r \int_0^1 \left[h \frac{\partial h}{\partial \eta} + \beta h^2 \frac{\partial h}{\partial \eta} \right] d\eta \tag{A.28}$$

$$= \alpha_r \int_0^1 \left[\frac{1}{2} \frac{\partial h^2}{\partial \eta} + \beta \frac{1}{3} \frac{\partial h^3}{\partial \eta} \right] d\eta \tag{A.29}$$

$$= \alpha_r \frac{1}{2} \left[h^2 \right]_0^1 + \frac{\alpha_r \beta}{3} \left[h^3 \right]_0^1 \tag{A.30}$$

$$= -\frac{\alpha_r h_m^2}{2} - \frac{\alpha_r \beta h_m^3}{3} \tag{A.31}$$

and from the equation (A.21)

$$H = -\int h dx = -\int h \Delta d\eta = \frac{\Delta}{4} h_m \left(1-\eta\right)^4 + \phi(t) \tag{A.32}$$

To determine the integration constant $\phi(t)$,

$$\begin{aligned}\left.\frac{\partial H}{\partial t}\right|_x &= \left.\frac{\partial H}{\partial t}\right|_\eta - \frac{1}{\Delta}\frac{\partial H}{\partial \eta}\left(\dot{S}_e + \eta\dot{\Delta}\right) \\ &= \frac{\dot{\Delta}}{4}h_m(1-\eta)^4 + \dot{\phi} + h_m\left(1-\eta\right)^4\left(\dot{S}_e + \eta\dot{\Delta}\right)\end{aligned} \tag{A.33}$$

At the boundary $x = S_e + \Delta$,

$$\left.\frac{\partial H}{\partial t}\right|_{S_e+\Delta} = 0 + \dot{\phi} + 0 = q(S_e + \Delta) = 0 \tag{A.34}$$

It means

$$\dot{\phi} = 0 \tag{A.35}$$

then

$$H = \frac{\Delta}{4}h_m(1-\eta)^4 + const. \tag{A.36}$$

also

$$\frac{\partial H}{\partial x} = \frac{1}{\Delta}\frac{\partial H}{\partial \eta} = -h_m(1-\eta)^3 \tag{A.37}$$

$$\left.\frac{\partial H}{\partial t}\right|_x = \frac{\dot{\Delta} h_m}{4}\left(1-\eta\right)^4 + h_m(1-\eta)^3\left(\dot{S}_e + \eta\dot{\Delta}\right) \tag{A.38}$$

Equation (A.23) finally can be integrated as

$$\int_{S_e}^{S_e+\Delta} \frac{\partial H}{\partial x}\frac{\partial H}{\partial t}\,dx \tag{A.39}$$

$$= \int_{S_e}^{S_e+\Delta} \left[-h_m(1-\eta)^3\right]\left[\frac{\dot{\Delta} h_m}{4}\left(1-\eta\right)^4 + h_m(1-\eta)^3\left(\dot{S}_e + \eta\dot{\Delta}\right)\right]dx \tag{A.40}$$

$$= -h_m^2 \int_0^1 (1-\eta)^3\left[\frac{\dot{\Delta}}{4}\left(1-\eta\right)^4 + (1-\eta)^3\left(\dot{S}_e + \eta\dot{\Delta}\right)\right]\Delta d\eta \tag{A.41}$$

$$= -h_m^2\,\Delta \int_0^1 \left[\frac{\dot{\Delta}}{4}\left(1-\eta\right)^7 + \dot{S}_e\left(1-\eta\right)^6 + \eta\left(1-\eta\right)^6\dot{\Delta}\right]d\eta \tag{A.42}$$

$$= -h_m^2\,\Delta \int_0^1 \left[\frac{\dot{\Delta}}{4}\left(1-\eta\right)^7 + \dot{S}_e\left(1-\eta\right)^6 + \eta\left(1-\eta\right)^6\dot{\Delta}\right]d\eta \tag{A.43}$$

$$= -h_m^2\,\Delta\left[\frac{1}{8}\frac{\dot{\Delta}}{4} + \frac{1}{7}\dot{S}_e + \frac{1}{56}\dot{\Delta}\right] \tag{A.44}$$

Now, we combine (A.31) and (A.44)

$$-\frac{\alpha_r h_m^2}{2} - \frac{\alpha_r \beta h_m^3}{3} = -h_m^2\,\Delta\left[\frac{11}{224}\dot{\Delta} + \frac{1}{7}\dot{S}_e\right] \tag{A.45}$$

$$\frac{11}{224}\dot{\Delta}+\frac{1}{7}\dot{S}_e=\frac{\alpha_r}{2}\left[\frac{1}{2}+\frac{\beta h_m}{3}\right] \tag{A.46}$$

Now, we have two equations and two unknowns for the state variables, the burn off rate ($\dot{S}_e$) and the boundary layer thickness (Δ).

$$\begin{bmatrix} \frac{11}{224} & \frac{1}{7} \\ \frac{1}{4} & \frac{h_{\text{sup}}}{h_m} \end{bmatrix}\begin{Bmatrix} \dot{\Delta} \\ \dot{S}_e \end{Bmatrix}=\begin{bmatrix} \frac{\alpha_r}{\Delta}\left(\frac{1}{2}+\frac{\beta h_m}{3}\right) \\ \frac{p_m}{h_m} \end{bmatrix} \tag{A.47}$$

Using (A.20) and (A.47) to solve for $\dot{\Delta}$ and $\dot{S}$ results in the desired state equations for the melting problem as

$$\dot{\Delta}=\frac{224(\Lambda^*+1)}{3\Lambda^*+11}\frac{\alpha_r}{\Delta}\left(\frac{1}{2}+\frac{\beta h_m}{3}\right)-\frac{32\Lambda^*}{3\Lambda^*+11}\frac{p_m}{h_m} \tag{A.48}$$

$$\dot{S}=-\frac{56\Lambda^*}{3\Lambda^*+11}\frac{\alpha_r}{\Delta}\left(\frac{1}{2}+\frac{\beta h_m}{3}\right)+\frac{11\Lambda^*}{3\Lambda^*+11}\frac{p_m}{h_m} \tag{A.49}$$

where, Λ^* is Stefan number.

$$\Lambda^*=\frac{h_m}{h_{\text{sup}}-h_m} \tag{A.50}$$

After simplifying the constant terms, the state equations for the electrode are

$$\dot{\Delta} = \frac{\alpha_r C_{\Delta\Delta}}{\Delta} - \frac{C_{\Delta p}}{h_m} p_m \tag{A.51}$$

$$\dot{S}_e = -\frac{\alpha_r C_{s\Delta}}{\Delta} + \frac{C_{sp}}{h_m} p_m \tag{A.52}$$

where, dimensionless constants are functions of volume specific enthalpy, Stefan number, and diffusivity coefficient.

$$C_{\Delta\Delta} = \frac{224(\Lambda^* + 1)}{3\Lambda^* + 11}\left(\frac{1}{2} + \frac{\beta h_m}{3}\right) \tag{A.53}$$

$$C_{\Delta p} = \frac{32\Lambda^*}{3\Lambda^* + 11} \tag{A.54}$$

$$C_{s\Delta} = \frac{56\Lambda^*}{3\Lambda^* + 11}\left(\frac{1}{2} + \frac{\beta h_m}{3}\right) \tag{A.55}$$

$$C_{sp} = \frac{11\Lambda^*}{3\Lambda^* + 11} \tag{A.56}$$

APPENDIX B

DETERMINATION OF THE VOLTAGE, RESISTANCE, AND IMMERSION DEPTH RELATIONSHIP BY EXPERIMENTAL DATA

The immersion depth is essentially the ESR equivalent to gap for the VAR. It considered to be the mean depth that the electrode is immersed into the molten slag, since the tip of the melting electrode is generally not flat. At present, there is no direct method for determining the actual immersion depth, so it must be inferred from a relationship with the electrical signals [34]. It is well known that the voltage and impedance of the ESR process increase with decreasing immersion depth. We will derive the relationship between voltage and the immersion depth and ultimately will find the resistance as a function of the immersion depth and the resistivity. Then we can relate the immersion depth with resistance. Basically, the slag in an ESR furnace acts as a resistor, whose resistance is determined by the usual formula:

$$R = \frac{V}{I} = \frac{\rho_e l(d)}{A_i} \tag{B.1}$$

where $\rho_e = \rho_0 e^{-CT_s}$ is resistivity and unit is (Ω-cm), $l(d)$ is a length of current path (*cm*), and $A_i = \pi r_i^2$, the area of ingot. In practice, the slag resistivity is difficult to quantify because process variables may change throughout the course of the melt that will alter its value. Some of those variables are slag chemistry, slag cap thickness, and slag temperature. Consequently, voltage swing or variance has been used as an indicator of immersion depth but unfortunately no relationship has been determined which relates swing to immersion depth. More detail about using voltage swing as a signature of the immersion depth control can be found from the literature [34]. A dedicated experiment for determining a more appropriate form of the relationship of the electrical signals to the immersion depth was conducted using 20.32 cm 1018 electrode into a 25.4 cm mold using 12 kg of 60/20/20 slag at the liquid metal processing laboratory at Sandia national laboratories. Several times, the electrode was withdrawn from the slag until the voltage hit 60 volts. The voltage and current were continuously recorded during these events yielding a continuous set of data whereby immersion depth and the electrical signals were related. Based on analysis of that data, an appropriate form of the immersion depth relationship turned out to be:

From the experimental data for E147 experiment, the temperature dependence of the slag resistance is determined as [Beaman Unpublished],

$$R = (-0.0197(T_s - 273) + 54.835) * 10^{-3} \tag{B.2}$$

At $T_s = 2073$ (K), the resistance was measured as $R_{1800} = 0.019375$ (Ω) and At $T_s = 2373$ (K), the resistance was measured as $R_{2100} = 0.013465$ (Ω). If we apply these results into the equation (B.1) then

$$R_{1800} = \frac{\rho_0 l(d_0) e^{-C(2073)}}{A_i}$$
$$R_{2100} = \frac{\rho_0 l(d_0) e^{-C(2373)}}{A_i} \tag{B.3}$$

then, we can determine C for the resistivity value of the slag,

$$\frac{R_{1800}}{R_{2100}} = 1.4389 = e^{-C(2073-2373)}$$
$$300C = \ln(1.4389) \tag{B.4}$$
$$\therefore C = 1.213*10^3 \,(1/K)$$

From the experiment, the current, the immersion depth, and the slag temperature were set and the resulting melt rate is about 25 (g/sec) and the voltage is around 27 (V).

$$I_* = 3000\,A$$
$$d_* = 0.5\,cm$$
$$T_{s*} = 2100\,K \tag{B.5}$$
$$V \cong 27\,V$$
$$\dot{m}_* = 25\,(g/\sec)$$

Then voltage can be expressed as

$$\begin{aligned} V &= R(d,\sigma(T_s))I \\ &= \frac{\rho_0 e^{-CTs} l(d)}{A_i} I \end{aligned} \tag{B.6}$$

From the experimental data (when, $I = 3000$ (A))

$$V = V_{2d} d^2 - V_{1d} d + V_{0d} \tag{B.7}$$

See, Figure B.1 for voltage and immersion depth relationship. By using linear regression, the parameters V_{2d}, V_{1d}, V_{0d} have been determined for E147 experiment.

$$\begin{aligned} V_{2d} &= 56.42 \text{ (V/cm}^2\text{)} \\ V_{1d} &= 88.43 \text{ (V/cm)} \\ V_{0d} &= 58.19 \text{ (V)} \end{aligned} \tag{B.8}$$

If we apply these parameters to the equation (B.5), then V_*=28.08 (V) . The equation (B.6) is

$$\rho_0 l(d_*) = A_i e^{CT_{s*}} \frac{V_*}{I_*} \tag{B.9}$$

Also

$$\begin{aligned} V(d, \sigma(T_{s*})) &= (V_{2d} d^2 - V_{1d} d + V_{0d}) \\ &= \frac{\rho_0 e^{-CT_{s*}} l(d)}{A_i} I_* \end{aligned} \tag{B.10}$$

To derive resistance relationship, the equation (B.10) can be restated as

$$\begin{aligned}\rho_0 l(d) &= \frac{A_i e^{CT_{s*}}}{I_*}\left(V_{2d}d^2 - V_{1d}d + V_{0d}\right) \\ &= A_i e^{CT_{s*}}\left(R_{2d}d^2 - R_{1d}d + R_{0d}\right)\end{aligned} \tag{B.11}$$

Where

$$\begin{aligned}R_{2d} &= \frac{V_{2d}}{I_*} \\ R_{1d} &= \frac{V_{1d}}{I_*} \\ R_{0d} &= \frac{V_{0d}}{I_*}\end{aligned} \tag{B.12}$$

Finally,

$$V = \left(R_{2d}d^2 - R_{1d}d + R_{0d}\right)e^{-C(T_s - T_{s*})}I \tag{B.13}$$

and the resistance as a function of immersion depth and the resistivity can be described as

$$R(d,\sigma(T_s)) = \left(R_{2d}d^2 - R_{1d}d + R_{0d}\right)e^{-C(T_s - T_{s*})} \tag{B.14}$$

Now let's relate this resistance information to the immersion depth. The derived equation will be used to formulate the measurement dynamics equation. While the electrode is immersed in the slag, the resistance can be expressed as (similar to the equation (B.1))

$$R = \frac{f(d)}{\sigma L} \qquad \text{(B.15)}$$

where, $\sigma = \sigma_0(x)e^{[A(T-T_*)]}$, x is concentration effects, L = characteristic length constant, and *f(d)* is dimensionless form factor. The simple diagram shows this idea is given in Figure B.2

Assume *f(d)* is linear with respect to immersion depth, *d*.

$$f(d) = f_0 - f_1(d - d_0) \qquad \text{(B.16)}$$

and f_0 and f_1 are constants, then

$$R = \frac{f_0 - f_1(d - d_0)}{\sigma(x,T)L} \qquad \text{(B.17)}$$

At $d = d_0$, $R = R_0 = \frac{f_0}{\sigma(x,T)L}, R_1 = \frac{f_1}{\sigma(x,T)L}$ is the slope. Put these relations to the equation (B.17) then we can get

$$R = R_0 - R_1(d - d_0) \qquad \text{(B.18)}$$

Finally, the immersion depth can be derived as

$$d = d_0 + \frac{R_0 - R}{R_1} \qquad \text{(B.19)}$$

the R_0, R_1 can be determined from the experiment.

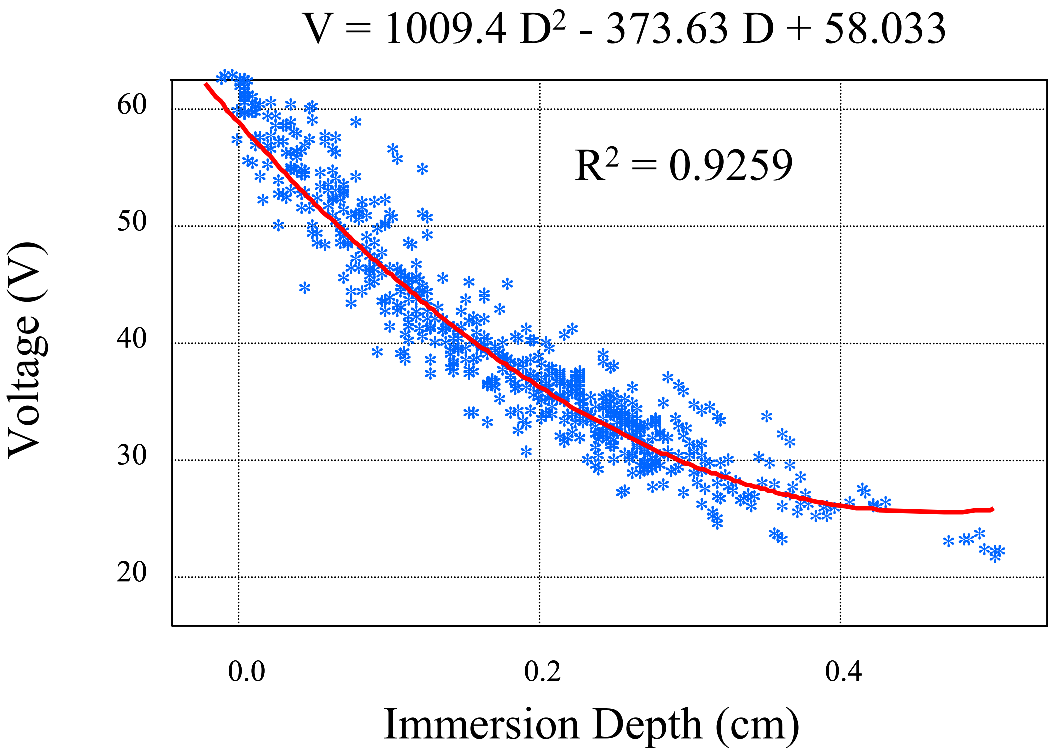


Figure B.1 Voltage vs. immersion depth relationship

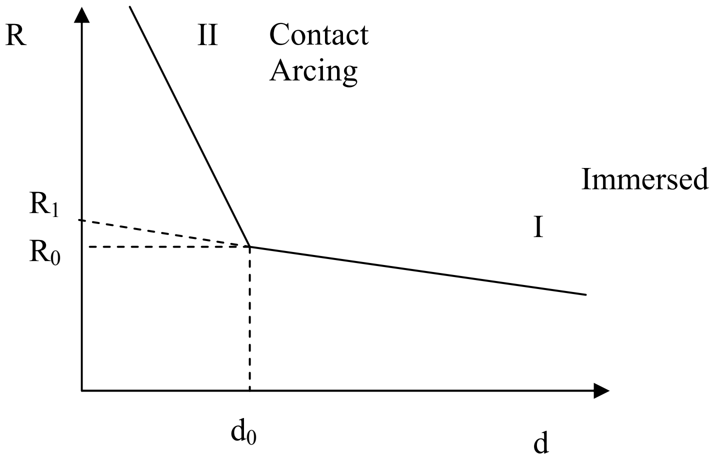


Figure B.2 Schematic diagram showing the immersion depth and resistance relationship

APPENDIX C

USER MANUAL FOR ESR CONTROL TOOLBOX VERSION 1.0

ESR Control Toolbox Version 1.0 is a Matlab® language simulation tool designed to help and guide the control design of electroslag remelting process. The unscented Kalman filter algorithm used in this toolbox is a modified version of Julier's original Matlab® code and van der Merwe's code (before ReBEL). The program is composed of

- *parameter.m*: include all the parameters for the simulation
- *fun_esr.m*: describes dynamics of the ESR process (discrete-time)
- *hfun_esr.m*: describes dynamics of the measurement of ESR process (discrete-time)
- *lkf_esr.m*: simulates control design of ESR process with linearized time-invariant Kalman filtering estimation algorithm
- *lkf*.m: calculates constant Kalman gain off-line for *lkf_esr.m*
- *ekf_esr.m*: simulates control design of ESR process with extended Kalman filtering estimation algorithm
- *ukf_esr*.m: simulates control design of ESR process with unscented Kalman filtering estimation algorithm
- *ukf_mod.m*: calculates sigma points and unscented Kalman gain

- *mc_esr.m*: simulates control design of ESR process with sequential importance sampling based particle filtering estimation algorithm

The specific parameters used for the simulation and the control parameters are summarized in Table 6.7 and 6.8.

PARAMETER_ESR.M

First, setup for parameter file: *parameter_esr.m* based on the following guideline.

1. Dimensional parameters
2. Electrode electrical parameters
3. Thermophysical properties for the electrode material
4. Thermophysical properties of slag
5. Physical properties of slag
6. Calculated parameters
7. Nominal values
8. Noise strengths
9. Parameters from Immersion depth, voltage relationship
10. Parameters from Immersion depth, voltage relationship

1. Input Dimensional Parameters

```
% Parameter file for running simulation and Kalman gain calculation

% Dimensional Parameters

De = 8*2.54;                          % electrode dia (cm)
re = De/2;                            % electrode radius (cm)
Di = 10*2.54;                 % ingot dia (cm)
ri = Di/2;                            % crucible radius (cm)
```

2. Input Electrode Electrical Parameters

```
% SNL Eletrical parameters

Aelect = 1.213e-3;      % Resistivity temperature coefficient (1/°K)
R1 = .005;              % Impedance at inflection (ohm)
m0 = 1/70;              % Negative impedance/cm d<0 (ohm/cm)
m1 = 1/1300;            % Negative impedance/cm d>0 (ohm/cm)
Tsstar = 2200;          % Nominal slag temperature for voltage fit (°K)
```

3. Input Thermophysical Properties for the electrode material

```
% Thermophysical Properties of AISI 1018 steel

Tr = 300;               % Room tempeature (°K)
rhor = 7.83;            % Room temperasture density (g/cm^3)
Cr = .434;              % Room temperature specific heat (J/g-°K)
Kroom = .639;   % Room temperature thermal conductivity (W/cm-°K)
alphar = Kroom/rhor/Cr; % Room temperature diffusivity (cm^2/sec)
Tm = 1783;              % Liquidus temperature (°K)
rhom = 7.40;            % Melt temperasture density (g/cm^3)
Cm = 1.168;             % Melt temperature specific heat (J/g-°K)
Km = .313;              % Melt temperature thermal conductivity (W/cm-°K)
alpham = Km/rhom/Cm;    % Melt temperature diffusivity (cm^2/sec)
hm= .5*(rhom*Cm+rhor*Cr)*(Tm-Tr); % Approximate melt temperature Specific
Enthalpy (J/cm^3)
Tsup = Tm + 100;        % Superheat temperature (°K)
L = 65*4.184;           % Latent heat (J/g)
hsup = .5*(rhom*Cm+rhor*Cr)*(Tsup-Tr)+rhom*L; % Approximate superheat
specific enthalpy (J/cm^3)
```

4. Input Thermophysical Properties of Slag

```
% Thermophysical Properties of slag

Cs0 = 1.47;             % Specific heat of slag (J/g-°K)
rhos = 2.55;            % Density of Slag (g/cm^3)
Ks = .0418;             % Thermal conductivity of slag (W/cm-°K)
Tss = 1773;             % Solidus temperature of slag (°K)
```

5. Input Physical Properties of Slag

```
% Physical Properties of slag

Ms = 125*454;           % slag mass (g)
Vs = Ms/rhos;           % Volume of Slag (cm^3)
```

6. Calculated parameters

```
% Calculated Parameters

Ae = pi*re^2;           % electrode area (cm^2)
Ai = pi*ri^2;           % ingot area (cm^2)

% Dimensionless parameters

betam = (alpham-alphar)/alphar;% diffusivity parameter
lambda = hm/(hsup-hm);                 % Stefan number
a0 = 1-Ae/Ai;                          % nominal area ratio

% Model parameters

den = 3*lambda+11;
Cdd = 224*(lambda+1)*(.5+betam/3)/den;
Cdp = 32*lambda/den;
Csd0 = 56*lambda*(.5+betam/3)/den;
Csp = 11*lambda/den;
```

7. Input and Calculate Nominal Values

```
% Nominal values;

mu = .55*(pi*(5*2.54)^2/Ai);   % Melt efficiency
mdot0 = 50;                    % Nominal melt rate (g/sec)
Pm0 =  mdot0*hsup/rhom; % Nominal melt power (W)
P0 =Pm0/mu;                    % Nominal total power (W)
pm0 = Pm0/Ae;           % Nominal melt power flux (W/cm^2)
Ts0 = 2200;                    % Nominal slag temperature (°K)
d0 = 0.001;                    % Nominal immersion depth (cm)
hs0 = Ae*d0/Ai+Vs/Ai;   % Initial slag height (cm)
He = pm0/(Ts0-Tm);             % Heat transfer coefficient from slag to
electrode (W/cm^2-¡K)
Hs = (P0-Pm0)/2/pi/ri/hs0/(Ts0-Tss); % Heat transfer coefficient from slag
to slag skin (W/cm^2-¡K)
R0 = (R1-m1*d0)*exp(-Aelect*(Ts0-Tsstar)); % Nominal resistance (ohm)
I0 = sqrt(P0/R0);                      % Nominal current (A)
V0 = R0*I0;                            % Nominal volage (V)
delta0 = (hm*alphar*Cdd)/(Cdp*pm0);    % Initial boundary layer depth (cm)
Sdot0 = mdot0/rhom/Ae;         % Initial burn off rate
Vram0 = a0*Sdot0;              % Ram Velocity (cm/sec)
mur0=0;                 % Nominal relative efficiency
```

8. Input Noise Strengths

```
% Measurement Noise strengths

sigmadV = .05;          % depth standard deviation (cm)
sigmaPos = .3;          % encoder standard deviation (cm)
sigmaImeas = 200;              % Hall effect current sensor standard
deviation(A)
sigmaLC = 500;          % Load cell standard deviation (gm)
sigmaV = .1;                   % Voltage standard deviation (V)

% Input noise strengths
sigmaI = 184.2;         % Current standard deviation (A)
sigmaVram = .01;        % Ram speed standard deviation (cm/sec)

% Process modeling noise strengths

sigmamur = .0005;       % Melt efficiency standard deviation
sigmaa = .01*a0;% Fill ratio parameter standard deviation
sigmaIb = .0005*I0;% Bias current standard deviation
sigmaVramb = .1*Vram0;  % Bias ram speed standard deviation
sigmaVoltb = .0005*V0;  % Bias voltage standard deviation
```

9. Input Parameters from Immersion Depth, Voltage Relationship

```
% Impedance coefficients

k0 = 1/m0;
k1 = 1/m1;

% Resistance values
R1 = .006;
R1e = .0065;

dInflection = 0;
```

10. Input Parameters from Immersion Depth, Voltage Relationship

```
% Control gains
% Slag temperature

        SlagTemperatureTimeConstant = 10;    % (sec)
        KTs = 1/SlagTemperatureTimeConstant;

% Immersion depth
        DepthControlTimeConstant = 1; % (sec)
        Kd = 1/DepthControlTimeConstant;
```

FFUN_ESR.M

ffun_ESR.m is a function file which calculates the updated states based on the model we derived in Chapter 5. Input is previous states, control inputs, noises, and sampling time. Output is updated states based on the model. The original codes can be obtained by the author and SMPC with an agreement.

```
function [y] = ffun_ESR(x,u,noise,ts);
% PURPOSE : Process model function for ESR process.
% INPUTS  : - x:  The evaluation point in the domain.
% OUTPUTS : - y:  The value of the function at x.
% AUTHORS : Seokyoung Ahn (syahn@mail.utexas.edu), Joseph J. Beaman
% DATE    : Sep 2003
.
.
.
```

HFUN_ESR.M

hfun_ESR.m is a function file which calculates the updated measurement states based on the model we derived in Chapter 5. Input is previous states, control inputs, noises, and sampling time. Output is current time measurement estimate based on the model. The original codes can be obtained by the author and SMPC with an agreement.

```
function [y] = hfun_ESR(x,u,noise,ts);
% PURPOSE : Measurement model function for ESR process.
% INPUTS  : - x: The evaluation point in the domain.
% OUTPUTS : - y: The value of the function at x.
% AUTHORS : Seokyoung Ahn(syahn@mail.utexas.edu), Joseph J. Beaman
% DATE    : Sep 2003
.
.
.
```

LKF.M

lkf.m is a function file which calculates a Kalman gain based on the linearized Kalman filtering algorithm at nominal states. This program calculates Jaconbians of the model and evaluates them with nominal values in discrete form. We used kalman.m Matlab® internal function to calculate Kalman gain.

```
% Initialization and Parameters
% parameter_mod;            % Load parameter file

% Measurement model Sensitivities
dRdd = -m1*exp(-Aelect*(Ts0-Tsstar));        %dR/dd (ohm/cm) (d0 >= 0)
dPdd = dRdd*I0^2;                            %dP/dd (W/cm)
dRdTs = -(R1-m1*d0)*Aelect*exp(-Aelect*(Ts0-Tsstar));%dR/dTs (ohm/¡K)
dPdTs = dRdTs*I0^2;                          %dP/dTs (W/¡K)
dpmdmur = He*(Ts0-Tm);                       %dpm/dmur (W/m^2)
dpmdTs = (1+mur0)*He;                        %dpm/dTs (W/m^2-¡K)
dVdI = R0;                                   %dV/dI (ohm)
dPdI = dVdI*I0 + V0;                         %dP/dI  (V)
dPdVb = I0;                                  %dP/dVb (A)

% Jacobian calculation for state variables
Amm = zeros(10,10);
% Delta coefficients
Amm(1,1) = -alphar*Cdd/delta0^2;
Amm(1,2) = -Cdp*dpmdTs/hm;
Amm(1,6) = -Cdp*dpmdmur/hm;
% Ts coefficients
Amm(2,2) = (dPdTs-He*Ae-2*pi*Hs*ri*hs0)/rhos/Vs/Cs0;
Amm(2,3) = dPdd/rhos/Vs/Cs0;
Amm(2,8) = dPdVb/rhos/Vs/Cs0;
Amm(2,9) = dPdI/rhos/Vs/Cs0;
% d coefficients
Amm(3,1) = -alphar*Csd0/delta0^2;
Amm(3,2) = -Csp*dpmdTs/hm;
Amm(3,6) = -Csp*dpmdmur/hm;
Amm(3,7) = -(1/a0^2)*Vram0;
Amm(3,10) = 1/a0;
% Xram coefficients
Amm(4,10) = 1;
% me coefficients
Amm(5,1) = rhom*Ae*alphar*Csd0/delta0^2;
Amm(5,2) = rhom*Ae*dpmdTs*Csp/hm;
Amm(5,6) = rhom*Ae*Csp*dpmdmur/hm;
```

```
% Input matrix
Bm = zeros(10,2);
Bm(2,1) = dPdI/rhos/Vs/Cs0;
Bm(3,2) = 1/a0;
Bm(4,2) = 1;

% Noise input matrix
Gm = zeros(10,7);
Gm(2,1) = dPdI/rhos/Vs/Cs0;
Gm(3,2) = 1/a0;
Gm(4,2) = 1;

Gm(6,3) = 1;
Gm(7,4) = 1;
Gm(8,5) = 1;
Gm(9,6) = 1;
Gm(10,7) = 1;

H = zeros(5,10);
H(1,3) = 1;
H(2,4) = 1;
H(3,9) = 1;
H(4,3) = -rhos*Ae;
H(4,5) = 1;
H(5,2) = dRdTs*I0;
H(5,3) = dRdd*I0;
H(5,8) = 1;
H(5,9) = R0;

Fu = [0 0;
     0 0;
     1 0;
     0 0;
    R0 0];

Fw = zeros(5,7);
Fw(3,1) = 1;
Fw(5,1) = R0;

% Discrete-time Model of dynamics
ts = 2/15;     % Sampling time

sysc=ss(Amm,[Bm Gm],H,[Fu Fw]);
sysd=c2d(sysc,ts);

Q = [(sigmaI)^2 0 0 0 0 0 0;
```

```
    0 (sigmaVram)^2 0 0 0 0 0;
    0 0 ts*sigmamur^2 0 0 0 0;
    0 0 0 ts*sigmaa^2 0 0 0;
    0 0 0 0 ts*sigmaVoltb^2 0 0;
    0 0 0 0 0 ts*sigmaIb^2 0;
    0 0 0 0 0 0 ts*sigmaVramb^2];       % process noise variance.

Rr = [sigmadV^2 0 0 0 0;
     0 sigmaPos^2 0 0 0;
     0 0 sigmaImeas^2 0 0;
     0 0 0 sigmaLC^2 0;
     0 0 0 0 sigmaV^2];          % measurement noise variance.

% Discrete-time Kalman filter calculation
[KEST,M,P,MI,Z] = kalman(sysd,Q,Rr);
```

****F_ESR.M**

lkf_ESR.m, ekf_ESR.m, ukf_ESR.m and MC_ESR.m programs share lots of common in initialization parts and calculation of control inputs parts. We will show a basic structure how they are programmed. The original codes can be obtained by the author and SMPC with an agreement.

1. Initialization and Parameters Setup
    - Load parameter_esr.m
    - Load Kalman gain (for lkf_esr.m case)
2. Scaling parameters for UKF related algorithm (for ukf_esr.m case)
    - The parameters for SUT, (α, β, κ)
3. Simulation setup variables
    - start time, end time, time step calculation
4. Initialization of states and covariances (vectorization for speed up)
    - Initialization of states and covariances
    - Initialization of estimated states and covariances

- Initialization of measurement states

5. Main loop for statistical average
    - Sub loop for time step evolution
        a. Noise generation for simulator
        b. Set melt rate profile
        c. covariance assigning from the previous step (EKF, UKF)
        d. Estimate states value assigning from previous time step
        e. Compute related variables and control inputs based on the estimates
6. This parts LKF, EKF, UKF, and MC are different
    a. LKF
        i. A priori estimate updates
        ii. Simulator variables updates and save for plotting
        iii. Assign updated estimates and updated simulator states
        iv. Measurement simulator states and measurement estimates generation based on the current simulator states and estimates
        v. Estimates update with Kalman gain
        vi. sub loop for time step ends
        vii. Calculate errors, RMSE, etc.
    b. EKF
        i. A priori estimate updates
        ii. Jacobian calculation for states with a priori estimate

   iii. A priori covariance updates
   iv. Simulator variables updates and save for plotting
   v. Assign updated estimates and updated simulator states
   vi. Measurement simulator states and measurement estimates generation based on the current simulator states and estimates
   vii. Calculates measurement Jacobians
   viii. Kalman gain calculation and states and covariance updates
   ix. sub loop for time step ends
   x. Calculates errors, RMSE, etc.

c. UKF
   i. Simulator variables updates and save for plotting
   ii. Assign updated simulator states
   iii. Measurement simulator states generation based on the current simulator states
   iv. Input current estimate, current covariance, control input, noise variances, scaling parameters for SUT into *ukf_mod.m*. Outputs are updated mean and covariances
   v. sub loop for time step ends
   vi. Calculates errors, RMSE, etc.

REFERENCES

[1] Alspach, D. L. and Sorenson, H .L. "Nonlinear Bayesian Estimation using Gaussian Sum Approximation," *IEEE Transactions on Automatic Control*, Vol. 17, No.4, 1972, pp. 439-448.

[2] Arurlampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, Vol. 50, No.2, 2002, pp. 174-188.

[3] Athans, M., Wishner, R. P., and Bertolini, A. "Suboptimal State Estimation for Continuous-Time Nonlinear Systems from Discrete Noisy Measurement," *IEEE Transactions on Automatic Control*, Vol. AC-13, No. 5, 1968, pp. 504-514.

[4] Beaman, J. J., Ahn, S., Tetzlaff, D. A., Van Den Avyle, J., and Melgaard, D. K. "Modeling and Control of the Electroslag Remelting Process," *Researcher's Conference Proceedings*, Amarillo National Resource Center for Plutonium, 1999, pp. 65-67.

[5] Beaman, J. J., Williamson, R. L., and Melgaard, D. K. "A Nonlinear Reduced Order Model for Dynamic VAR Melt Rate Control," *Proceedings of the International Symposium on Liquid Metal Processing and Casting*, Santa Fe, New Mexico, 2001, pp. 161-174.

[6] Beaman, J. J., Williamson, R. L., and Melgaard, D. K., Shelmidine, G. J., and Hamel, J. C. "Model Based Gap and Melt Rate Control for VAR of Ti-6Al-4V," *Proceedings of the International Symposium on Liquid Metal Processing and Casting*, Nancy, France, 2003, pp. 67-76.

[7] Beněs, V. E. "Exact Finite-dimensional Filters with Certain Diffusion Nonlinear Drift," *Stochastics*, Vol. 5, 1981, pp 65-92.

[8] Bertram, L. A. and Zanner, F. J. "Electrode Tip Melting Simulation during Vacuum Arc Remelting of Inconel 718," *Modeling and Control of Casting and Welding Processes*, ed. TMS, Warendale, PA, 1986, pp. 95-106.

[9] Campbell, M. E. and Brunke, S. “Nonlinear Estimation of Aircraft Models for On-line Control Customization,” *IEEE Aerospace Conference Proceedings*, Vol. 2, 2001, pp. 2621-2628.

[10] Daum, F. E. “Exact Finite Dimensional Filters,” *IEEE Transactions on Automatic Control*, Vol. AC-31, No. 7, 1986, pp. 616-622.

[11] Daum, F. E. and Huang J. “Curse of Dimensionality and Particle Filter,” *Proceedings in IEEE Aerospace Conference*, Vol. 4, 2003, pp. 1979-1993.

[12] Doucet, A., de Freitas, N., and Gordon, N. Sequential Monte Carlo Methods in Practice, Springer-Verlag, April 2001.

[13] Duckworth, W. E. and Hoyle, G. Electroslag Refining, Chapman and Hall Ltd., London, 1969.

[14] Gelb, A. Applied Optimal Estimation, MIT Press, 1974.

[15] Godsill, S., Doucet, A. and West, M. “Methodology for Monte Carlo Smoothing with Application to Time-varying Auto regressions,” *Proceedings of International Symposium on Frontiers of Time Series Modeling*, 2000.

[16] Gordon, N. J., Salmond, D. J., and Smith, A. F. M. “Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation,” *IEE Proceedings-F* Vol. 140, No. 2, April 1993, pp. 107-113.

[17] Handschin, J. E. “Monte Carlo Techniques for Prediction and Filtering of Non-linear Stochastic Processes,” *Automatica*, Vol.6, 1970, pp. 55-563.

[18] Handschin, J. E. and Mayne, D. Q. “Monte Carlo Techniques to Estimate the Conditional Expectation in Multi-Stage Non-linear Filtering,” *International Journal of Control*, Vol. 9, No. 5, 1969, pp. 547-559.

[19] Herman S. M. A Particle Filtering Approach to Joint Passive Radar Tracking and Target Classification, PhD Thesis, The University of Illinois at Urbana-Champaign, 2002.

[20] Hoyle G. Electroslag Process, Applied Science Publishers, Ripple Road, Barking, Essex, England, 1983.

[21] Ito, K., Xiong, K. "Gaussian Filters for Nonlinear Filtering Problems," *IEEE Transactions on Automatic Control*, Vol.45, No. 5, 2000, pp. 920-927.

[22] Jazwinski, A. H. Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.

[23] Julier, S. J., Uhlmann, J., and Durrant-Whyte, H. "A New Approach for Filtering Nonlinear Systems," *Proceedings of the American Control Conference*, 1995, pp. 1628-1632.

[24] Julier, S. J. "The Scaled Unscented Transformation," *Proceedings of the American Control Conference*, 2002, pp. 4555-4559.

[25] Julier, S. J. and Uhlmann, J. K. "A General Method for Approximating Non-linear Transformations of Probability Distributions," *Tech. Rep., RRG*, Dept. of Engineering Science, University of Oxford, 1996, http://citeseer.nj.nec.com/julier96general.html .

[26] Julier, S. J. and Uhlmann, J. K. "A New Extension of the Kalman Filter to Nonlinear Systems," *Proceedings of SPIE*, Vol. 3068, 1997, pp. 182-193.

[27] Julier, S. J. and Uhlmann, J. K. "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, 2004, pp. 401-422.

[28] Kalman, R. E. "A New Approach to Linear Filtering and Prediction Problems," *ASME Journal of Basic Engineering*, Vol. 82 (Series D) , 1960, pp. 35-45.

[29] Kong, A., Liu, J. S., and Wong, W. H. "Sequential Imputations and Bayesian Missing Data Problems," *Journal of the American Statistical Association*, Vol. 89, No. 425, 1994, pp. 278-288.

[30] Lefebvre, T., Bruyninckx, H., and De Schutter, J. "Comment on "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators"," *IEEE Transactions on Automatic Control*, Vol. 47, No.8, 2002, pp. 1406-1409.

[31] Maybeck, P. S. Stochastic Models. Estimation and Control, Vol.2 of Mathematics in Science and Engineering. Navtech Books & Software, 1994.

[32] Melgaard, D. K., Williamson, R. L., and Beaman, J. J. "Controlling Remelting Processes for Superalloys and Aerospace Ti Alloys," *JOM*, 1998, pp. 13-17.

[33] Melgaard, D. K. and Shelmidine, G. J. "ESR process Instabilities while Melting Pipe Electrodess," *Proceedings of the International Symposium on Liquid Metal Processing and Casting*, Santa Fe, New Mexico, 1999, pp. 176-186.

[34] Melgaard, D. K. and Shelmidine, G. J. "Electroslag Remelting Immersion Depth Control Using Characteristic Voltage Signatures," *Proceedings of the International Symposium on Liquid Metal Processing and Casting*, Santa Fe, New Mexico, 2001, pp. 148-160.

[35] Nørgaard, M., Poulsen, N., and Ravn, O. "New Development in State Estimation for Nonlinear Systems," *Automatica*, Vol. 36, No. 11, 2000, pp. 1627-1638.

[36] Ohmura, T., Wakabayashi M., and Hosoda, T. "A Heat Transfer Analysis of ESR Process," *5th Annual ESR Symposium Conference*, 1983.

[37] Pitt, M. and Shephard, N. "Filtering via Simulation: Auxiliary Partile Filters," *Journal of the American Statistical Association*, Vol. 94, No. 446, 1999, pp. 590-599.

[38] Rabiner, L. R. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of IEEE*, Vol. 77, 1989, pp. 257-285.

[39] Ristic, B., Arulampalam, A., and Gordon, N. Beyond the Kalman Filter, Artech House Publishers, Boston, 2004.

[40] Scholte, E. and Campbell, M. E. "On-line Nonlinear Guaranteed Estimation with Application to a High Performance Aircraft," *Proceedings of the American Control Conference*, 2002, pp. 184-190.

[41] Tenne, D. Statistics based Sampling for Controller and Estimator Design, Dissertation, State University of New York at Buffalo, Buffalo, New York, 2004.

[42] Tetzlaff, D. A. Understanding the Physical Phenomena in the Electro Slag Remelting Process, Master Thesis, The University of Texas at Austin, Austin, Texas, 2000.

[43] Tweten, D. J. Thermal and Electrical Model of the Electrode of the Electroslag Remelting Process, Master Thesis, The University of Texas at Austin, Austin, Texas, 2004.

[44] van der Merwe, R., de Freitas, N., Doucet, A., and Wan, E. “The Unscented Particle Filter,” *Tech. Rep. CUED/F-INFENG/TR 380*, Cambridge University Engineering Department, 2000.

[45] van der Merwe, R. and Wan, E. “Efficient Derivative-Free Kalman Filters for Online Learning,” *Proceedings of the 9th European Symposium on Artificial Neural Networks*, 2001, pp. 205-210.

[46] van der Merwe, R. Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models, PhD Thesis, Oregon Health and Science University, 2004.

[47] Vujanović, B. D. and Jones, S. E. “Approximate Solutions of Canonical Heat Conduction Equations,” *Transactions of ASME, Journal of Heat Transfer*, Vol. 112, 1990, pp. 836-842.

[48] Wan, E. and van der Merwe, R. Kalman Filtering and Neural Networks, Ed. Simon Haykin, Chapter 7 “The Unscented Kalman Filter,” Wiley, 2001.

[49] Williamson, R. L., Beaman, J. J., Melgaard, D. K., Shelmidine, G. J., Tubesing, P. K., and Aikin, R. M. “Dynamic Melt Rate Control on a Laboratory Scale VAR Furnace without Load Cell Feedback,” *Proceedings of the International Symposium on Liquid Metal Processing and Casting*, Santa Fe, New Mexico, 2001, pp. 175-186.

[50] Williamson, R. L., Beaman, J. J., Melgaard, D. K., Shelmidine, G. J., Patel, A. D., and Adasczik, C. B. “A Demonstration of Melt Rate Control during VAR of “Cracked” Electrodes,” *Proceedings of the International Symposium on Liquid Metal Processing and Casting*, Nancy, France, 2003, pp. 29-38.

[51] Williamson, R. L., Beaman, J. J., Melgaard, D. K., Shelmidine, G. J., and Morrison, R. “Model-Based Melt Rate Control during Vacuum Arc Remelting of Alloy 718,” *Metallurgical and Materials Transactions B*, Vol. 35B, 2004. pp. 101-113.

[52] Van Zandt J. R. “A more robust unscented transform,” Technical report, MITRE Corporation, July 2001.

[53] Zien, T. “Approximate Calculation of Transient Heat Conduction,” *AIAA Journal*, Vol. 14, No. 3, 1976, pp.404-406.

[54] Zien, T. “Integral Solutions of Ablation Problems with Time-Dependant Heat,” *AIAA Journal*, Vol. 16, No. 12, 1978, pp. 1287-1295.

VITA

Seokyoung Ahn was born in Busan, South Korea on April 23th, 1970, the son of Jong-In Ahn and Mihea Koo. He entered the undergraduate program in Mechanical Engineering at Busan National University, Busan, South Korea in March 1988 and received the degree of Bachelor of Engineering in February 1992. He entered the graduate program in Mechanical Engineering at Pohang University of Science and Technology, Pohang, South Korea in March 1992 and received the degree of the Master of Science in February 1994. From April 1994 to June 1997, he worked as a full-time instructor in the department of Mechanical Engineering at the Naval Academy, Chinhae, South Korea as his military duty. He worked as an internet business consultant for EZNET corporation, Chinhae, South Korea till July 1998. He entered the Doctor of Philosophy in Mechanical Engineering at The University of Texas at Austin in August 1998.

Permanent address: 2501 Lake Austin Blvd F207, Austin, TX 78703
This dissertation was typed by the author.